

FINAL TOPICAL REPORT

“Multi-Dimensional Analysis of the Forced Bubble Dynamics Associated with Bubble Fusion Phenomena” (USDOE-NEER GRANT, DOE-FG02-99ID13776)

Richard T. Lahey Jr., Kenneth E. Jansen and Sunitha Nagrath

Center for Multiphase Research

and

Department of Mechanical, Aeronautical and Nuclear Engineering

Rensselaer Polytechnic Institute

Troy, New York, 12180-3590

December 2, 2002

Contents

1	Introduction	3
1.1	Overview	5
2	The Level Set Method	6
2.1	Interface thickness	7
2.2	Re-initialization or re-distancing of level sets	8
3	The Finite Element Formulation of the Flow	9
4	The Discontinuity Capturing Operator	13
5	Finite Element Formulation of the Level Set Method	14
5.1	Finite element discretization	14
5.2	Volume constraint	15
6	Modeling of Surface Tension	16
7	Ghost Fluid Method	17
8	Adaptive Mesh Strategy	20
8.1	Error indicators implemented	20
8.2	Refinement strategy	22
8.2.1	Edge marking	22
8.2.2	Mesh quality	23
9	Numerical Results and Discussion	23
9.1	Simulation of Incompressible multiphase flows	23
9.1.1	Simple advection of a cylindrical bubble	23

9.1.2	Two dimensional Simulation of bubble motion under buoyancy force at a medium Reynolds Number	26
9.1.3	Simulation of a three dimensional spherical bubble motion under buoy- ancy force	31
9.2	Simulation of Compressible flows	35
9.2.1	Steady shock problem	35
9.2.2	Sod's shock tube problem	39
9.2.3	Implosion of a spherical shock wave using an adaptive mesh strategy	40
9.2.4	Rayleigh-Taylor Instability	47
9.2.5	Imploding Air Bubble in Water	53
10	The Analysis of Bubble Fusion	53
11	Conclusions	57
12	Acknowledgements	58

1 Introduction

Sonoluminescence is a very complicated phenomena in which one or more small bubbles are trapped in a standing accoustic wave and then excited to a resonant condition. Under the right conditions, the resonance can be strong enough to cause the bubble(s) to implode creating ultra-high pressures and temperatures, creating light pulses (i.e, sonoluminescence). Understanding the stability of these bubbles is critical if they are to obtain the conditions sufficient for Bubble Fusion to occur. To this end, we have developed a three-dimensional direct numerical simulation (DNS) code that may be used for the analysis of both single and multiple bubble dynamics. This code, PHASTA-2C, efficiently represents the fluid dynamics of both phases (liquid and gas/vapor) and the interface between them. It is also capable of resolving the various shock waves which may be created during bubble implosions. Thus PHASTA-2C may be used as a HYDRO code (i.e, hydrodynamic shock code).

Existing computational methods used to solve two-phase flow problems, include front tracking methods [1], boundary integral methods [2], volume of fluid methods [3], phase field methods, front capturing methods, and level set methods [4, 5, 6]. All the above mentioned methods have their own advantages and disadvantages. An advantage of front tracking methods is with little resolution we can maintain high accuracy. However, significant remeshing is needed to prevent the marker particles (which track interface) from coming together at points of large curvature. Also the algorithms have to be modified to reconnect, or disconnect the free surface separating the various fluids. Volume of fluid methods are based on solving the conservation law for the volume fraction and they have excellent conservation properties within each phase. However a disadvantage of the volume of fluid method is that it is difficult to calculate the curvature of the front from the volume fractions. Furthermore, its extension to compresible flow sacrifices some of the aforementioned conservation properties. Although the level set method does not possess the same inherent conservation properties as volume of fluid methods or front tracking techniques, the strength of this method lies in it's ability to efficiently represent an arbitrarily complex interface very accurately, thereby allowing the computation of flows with surface tension and rapidly changing topology. Thus, in PHASTA-2C the interface between the two phases is captured using the level set approach

developed by Osher, Sethian and Sussman *et al.* [4, 5, 6], and the flow in each phase is represented by the Navier Stokes equations.

The Streamline-Upwind/Petrov-Galerkin method (SUPG) introduced by Hughes and Brooks, *et al.* [7] is used to discretize the governing flow and level set equations. As discussed by Hughes *et al.* [8], SUPG is an excellent method for problems with smooth solutions, but typically introduces localized oscillations about sharp internal and boundary layers. To improve this a discontinuity-capturing term proposed by Hughes *et al.* [9] is added to the formulation. This term provides additional control over gradients in the discrete solution and considerably increases the robustness of the methodology. The continuum surface force (CSF) model proposed by Brackbill *et al.* [10] is implemented in order to incorporate the surface tension stresses. Also an improved re-distancing strategy is adopted along with the discontinuity capturing to ensure the volume of the each phase is conserved while re-distancing. The method developed enables us to accurately compute the flows with large density and viscosity difference, as well as surface tension. In addition, the present formulation allows the fronts to intersect, merge, break, and change topology.

Although the resulting algorithm predicts single phase compressible gas dynamics well, for the compressible multiphase flows, it can lead to the spurious non-physical oscillations across the material interface due to a smeared density profile. This can be easily attributed to the radical change in the equation of state across the material front. So a numerical approach similar to the “Ghost Fluid Method” developed by Fedkiw and Aslam *et al.* [4, 5, 6] has been implemented. This strategy allows us to treat compressible multi-phase flows with large density difference. The phenomena of sonoluminescence presents us with a challenging problem, since it involves multiple shocks propagating with various strengths in each phase, regions with discontinuities, in addition to a rapidly changing interface. To study the problem efficiently, an adaptive mesh strategy is required. Hence the above discussed method has been implemented with an adaptive meshing strategy.

Several numerical studies have been performed to test PHASTA-2C. The studies are necessary exercises to reach the goal of developing a robust numerical method. Various incompressible problems were considered to assess code performance. Concomitantly, they

can be used as a testing tool for solving the multiphase flows using the level set method. In tandem they provide a check on the ability of the algorithm to treat deforming interfaces. The incompressible test cases include the simple advection of a gas bubble in liquid, the rising of a single gas bubble under buoyancy, a three dimensional simulation of the formation of a toroidal bubble, and finally the coalescence of two bubbles. Compressible analyses were performed on a wide variety of problems. Initially the performance of the code was tested against standard test problems to validate the predictions of important gas dynamics phenomena. These includes, Sod's shock tube problem, and a one dimensional standing shock problem. A single phase converging spherical shock problem was studied to validate the prediction of the shock waves and also the efficiency gained through the adaptive mesh strategy has been demonstrated. The well known Rayleigh-Taylor instability problem was analyzed as a test case for the computation of two-phase compressible dynamics with interface propagation. The wide and general (instead of a problem oriented) applicability of the developed algorithm was demonstrated through the study of these various two-phase problems. The DNS simulation of an imploding bubble dynamics was studied.

1.1 Overview

The outline of the report is as follows. In Section 2 we introduce the level set method. Section 3 deals with the finite element formulations for solving the compressible Navier-Stokes equations, in which we discretize in space using a stabilized finite element method (FEM) to obtain a nonlinear system of coupled ordinary differential equations. In Section 4, we describe the discontinuity capturing operator which was implemented to efficiently perform the flow computations with sharp discontinuities or involving shocks. Section 5 presents the details of the level set method and the finite element solution technique for solving both the level set and the re-distancing equations. Section 6 deals with the modeling of the surface tension force across the interface. To eliminate the spurious oscillations across the interface in compressible flows a novel approach, based on Ghost Fluid methods, has been adapted, and this is discussed in the Section 7. We discuss briefly the adaptive strategy used, in Section 8, and in Section 9 we present the results obtained from the simulations. This section

has two parts. The first part containing the results from the incompressible simulations and the second part containing the compressible simulations. In Section 10, a simulation of bubble fusion is presented, and shown to be in good agreement with recent ORNL data. Finally, in Section 11 some conclusions are drawn and future goals are presented.

2 The Level Set Method

The level set approach represents the free surface as a zero level set of a smooth function and simultaneously maintains a level set function which is the signed distance from the interface. Hence instead of explicitly tracking the interface, we implicitly “capture” the interface within a field which is interpolated between the finite element vertices like any other solution quantity (e.g pressure, velocity, temperature). This enables us to represent the interface between the two phases accurately and to compute flows with high density ratios and surface tension. As mentioned in [5], conventional conservative methods suffer excessive numerical diffusion which smears the sharpness of the front. The level set function is typically a smooth (Lipschitz continuous) function, denoted as ϕ , which eliminates the difficulties that conventional conservative schemes incur.

Probably the most important advantage of level set methods is that the interface can merge or break-up with no special attention. Furthermore, the level set formulation generalizes easily to three dimensions. The actual location of the interface is never computed, since the interface is embedded as a particular level set in a fixed domain.

A smooth level set function, ϕ , is used to track the interface between the gas phase and the liquid phase. In our formulations the interface, β , is the zero level set of ϕ :

$$\beta = \{\mathbf{x} | \phi(\mathbf{x}, t) = 0\} \quad (1)$$

The level set function is positive in the liquid phase and taken to be negative in the gas phase. Hence we have,

$$\phi(\mathbf{x}, t) = \left\{ \begin{array}{l} > 0, \text{ if } \mathbf{x} \in \text{the liquid} , \\ = 0, \text{ if } \mathbf{x} \in \beta, \\ < 0, \text{ if } \mathbf{x} \in \text{the gas.} \end{array} \right\} \quad (2)$$

Hence we initialize ϕ to be the signed normal distance from the interface. Since the interface moves with the fluid, the evolution of ϕ is governed by a transport equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \quad (3)$$

As discussed in Section 3, this additional advection equation for the level set scalar is solved in a manner similar to the flow equations. The physical properties of the fluid in each phase are calculated as a function of ϕ as given below

$$\rho(\phi) = \rho_1 H(\phi) + \rho_2 (1 - H(\phi)), \text{density} \quad (4)$$

and similarly,

$$\mu(\phi) = \mu_1 H(\phi) + \mu_2 (1 - H(\phi)), \text{viscosity} \quad (5)$$

where $H(\phi)$ is the Heaviside function given by

$$H(\phi) = \begin{cases} 0, & \text{if } \phi < 0, \\ \frac{1}{2}, & \text{if } \phi = 0, \\ 1, & \text{if } \phi > 0. \end{cases} \quad (6)$$

2.1 Interface thickness

Use of the Heaviside function described above leads to poor results due to the assumed zero thickness of the interface. Instead, it was replaced with an alternative description of the interface as proposed by [5, 1]. Numerically, they substitute the smoothed Heaviside function $H_\epsilon(\phi)$ for the sharp Heaviside function $H(\phi)$. The smooth Heaviside function is defined as [6],

$$H_\epsilon(\phi) = \begin{cases} 0, & \text{if } \phi < -\epsilon, \\ \frac{1}{2} \left[1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\phi}{\epsilon}\right) \right] & \text{if } |\phi| \leq \epsilon, \\ 1, & \text{if } \phi > \epsilon. \end{cases} \quad (7)$$

Where ϕ represents the signed normal distance to the free surface. The $\frac{1}{2}$ contour of the sharp Heaviside function $H(\phi)$ would have created jagged or staircase contours on any discrete mesh. Instead, by giving the interface a thickness of $\epsilon = \alpha \Delta x$ where $\alpha > 1$, sharp changes across the interface are smoothed.

2.2 Re-initialization or re-distancing of level sets

In the formulation describe above, the front will have a uniform thickness so long as ϕ is maintained as a distance function. But, under the evolution of equation (3), the level sets that are adjacent to the zero level set move with velocities different than that of a zero level set. Therefore, the ϕ distance field gets distorted. In this case, one must re-initialize the level set function on regular intervals in order to rebuild/maintain this signed distance function. There are several ways to accomplish this re-distancing. We followed the technique introduced by Sussman *et al.* [5]. Its virtue is that the level set function is re-initialized without explicitly finding the zero level set. The idea is to solve the partial differential equation

$$\frac{\partial d}{\partial \tau} = S(\phi)(1 - |\nabla d|), \quad (8)$$

where

$$S(\phi) = \begin{cases} -1, & \text{if } \phi < 0, \\ 0, & \text{if } \phi = 0, \\ 1, & \text{if } \phi > 0. \end{cases}, \quad (9)$$

$d(\mathbf{x}, 0) = \phi(\mathbf{x}, t)$ and τ is a pseudo time. Given any initial data for ϕ , solving this equation to steady state provides the re-distance distance field ϕ with the property $|\nabla \phi| = 1$, since the convergence occurs when the right hand side is zero. The sign function $S(\phi)$ controls the flow of information. If ϕ is negative, information flows one way, and if ϕ is positive, information flows the other way. The net effect is to straighten out the level sets on either side of the zero level set. The steady solutions of Eq(8) are distance functions. Furthermore, since $S(0) = 0$, then $d(\mathbf{x}, \tau)$ has the same zero level set as $\phi(\mathbf{x}, t)$. Note, that this equation is relaxed in pseudo-time τ which is not related to the physical time t . Hence we only need to solve Eq(8) for $\tau = 0 \dots \epsilon$, because the level set function re-initialization is required only near the front. This is obvious if we re-write the Eq(8) as

$$\frac{\partial d}{\partial \tau} + \mathbf{w} \cdot \nabla d = S(\phi), \quad (10)$$

where,

$$\mathbf{w} = S(\phi) \frac{\nabla d}{|\nabla d|} \quad (11)$$

The equation (10) is a non-linear hyperbolic equation with the characteristics pointing outwards from the interface in the direction of the normal. The strategy adopted in PHASTA-2C is to perform this re-distancing operation at the end of each time step.

3 The Finite Element Formulation of the Flow

Consider the compressible Navier-Stokes equations (complete with continuity and total energy equations) written in conservative matrix form (see [11, 12] for details):

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i}^{\text{adv}} - \mathbf{F}_{i,i}^{\text{diff}} = \mathcal{S} \quad (12)$$

where,

$$\mathbf{U} = \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{Bmatrix} = \rho \begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e_{tot} \end{Bmatrix}, \quad (13)$$

$$\mathbf{F}_i^{\text{adv}} = u_i \mathbf{U} + p \begin{Bmatrix} 0 \\ \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \\ u_i \end{Bmatrix}, \quad (14)$$

$$\mathbf{F}_i^{\text{diff}} = \begin{Bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \tau_{ij} \ u_j - q_i \end{Bmatrix} \quad (15)$$

$$(16)$$

and,

$$\tau_{ij} = 2(\mu + \mu_T)(S_{ij}(\mathbf{u}) - \frac{1}{3}S_{kk}(\mathbf{u})\delta_{ij}) \quad , \quad S_{ij}(\mathbf{u}) = \frac{u_{i,j} + u_{j,i}}{2} \quad (17)$$

$$q_i = -(\kappa + \kappa_T)T_{,i} \quad , \quad \kappa_T = c_p \frac{\mu_T}{Pr_T} \quad , \quad e_{\text{tot}} = e + \frac{u_i u_i}{2} \quad , \quad e = c_v T \quad (18)$$

The variables are: the velocity u_i , the pressure p , the density ρ , the temperature T and the total energy, e_{tot} . The constitutive laws relate the stress, τ_{ij} , to the deviatoric portion of the strain, $S_{ij}^d = S_{ij} - \frac{1}{3}S_{kk}\delta_{ij}$, through a molecular viscosity, μ , plus turbulent viscosity, μ_T . Similarly, the heat flux, q_i , is proportional to the gradient of temperature with the proportionality constant given by the addition of a molecular conductivity, κ , and a turbulent conductivity, κ_T which is assumed proportional to the turbulent viscosity as described above. The formulations presented in this work can accommodate a general divariant fluid. A variety of equations of state can be considered; for example, the gas phase may be assumed to be governed by the ideal gas and the liquid phase was governed by a simple linear acoustic equation of state. Note that here we do not consider two fluids (with two viscosities, two conductivities and two constitutive laws) but instead the level set field ϕ will allow us to consider these fields as one global flow field for each varying state variables. Finally \mathcal{S} is a body force (or source) term, such as gravity and the force due to the surface tension. The representation of the surface tension surface force as a body force applied on the interface is dealt in detail in Section 6.

For the specification of the methods that follow, it is helpful to define the quasi-linear operator (with respect to variable vector, \mathbf{Y}) related to Eq(12) as,

$$\mathcal{L} \equiv \mathbf{A}_0 \frac{\partial}{\partial t} + \mathbf{A}_i \frac{\partial}{\partial x_i} - \frac{\partial}{\partial x_i} (\mathbf{K}_{ij} \frac{\partial}{\partial x_j}) \quad (19)$$

which can be naturally decomposed into time, advective, and diffusive portions

$$\mathcal{L} = \mathcal{L}_t + \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{diff}}. \quad (20)$$

Here $\mathbf{A}_i = \mathbf{F}_{i,\mathbf{Y}}^{\text{adv}}$ is the i^{th} Euler Jacobian matrix, \mathbf{K}_{ij} is the diffusivity matrix, defined such that $\mathbf{K}_{ij}\mathbf{Y}_{,j} = \mathbf{F}_i^{\text{diff}}$, and $\mathbf{A}_0 = \mathbf{U}_{,\mathbf{Y}}$ is the change of variables metric. For a complete description of $\mathbf{A}_0, \mathbf{A}_i$ and \mathbf{K}_{ij} , the reader is referred to [13, 14]. Using this, one can write (12) as simply $\mathcal{L}\mathbf{Y} = \mathcal{S}$.

To proceed with the finite element discretization of the Navier-Stokes Eqs(12), we must define the finite element approximation spaces. First, let $\bar{\Omega} \subset \mathbf{R}^N$ represent the closure

of the physical spatial domain (i.e. $\Omega \cup \Gamma$ where Γ is the boundary) in N dimensions; only $N = 3$ is considered here. The boundary is decomposed into portions with natural boundary conditions, Γ_h , and essential boundary conditions, Γ_g , i.e., $\Gamma = \Gamma_g \cup \Gamma_h$. In addition, $H^1(\Omega)$ represents the usual Sobolev space of functions with square-integrable values and derivatives on Ω .

Next, Ω is discretized into n_{el} finite elements, Ω^e . With this, define the trial solution space for the semi-discrete formulations as

$$\mathbf{V}_h = \{\mathbf{v} | \mathbf{v}(\cdot, t) \in H^1(\Omega)^m, t \in [0, T], \mathbf{v}|_{x \in \Omega^e} \in P_k(\Omega^e)^m, \mathbf{v}(\cdot, t) = \mathbf{g} \text{ on } \Gamma_g\}, \quad (21)$$

and the weight function space

$$\mathbf{W}_h = \{\mathbf{w} | \mathbf{w}(\cdot, t) \in H^1(\Omega)^m, t \in [0, T], \mathbf{w}|_{x \in \Omega^e} \in P_k(\Omega^e)^m, \mathbf{w}(\cdot, t) = \mathbf{0} \text{ on } \Gamma_g\}, \quad (22)$$

where $P_k(\Omega^e)$, is the space of all polynomials defined on Ω^e , complete to order $k \geq 1$, and m is the number of degrees of freedom ($m = 5$).

To derive the so called weak form of Eq(12), the entire equation is dotted with a vector of weight functions, $\mathbf{W} \in \mathbf{W}_h$, and integrated over the spatial domain. Integration by parts is then performed to move the spatial derivatives onto the weight functions thus decreasing the continuity requirements. This process leads to the integral equation (often referred to as the weak form): find $\mathbf{Y} \in \mathbf{V}_h$ such that

$$\begin{aligned} \int_{\Omega} (\mathbf{W} \cdot \mathbf{A}_0 \mathbf{Y}_{,t} - \mathbf{W}_{,i} \cdot \mathbf{F}_i^{\text{adv}} + \mathbf{W}_{,i} \cdot \mathbf{F}_i^{\text{diff}} + \mathbf{W} \cdot \mathbf{S}) d\Omega \\ - \int_{\Gamma} \mathbf{W} \cdot (-\mathbf{F}_i^{\text{adv}} + \mathbf{F}_i^{\text{diff}}) n_i d\Gamma \\ + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \mathcal{L}^T \mathbf{W} \cdot \boldsymbol{\tau} (\mathcal{L} \mathbf{Y} - \mathbf{S}) d\Omega = 0 \end{aligned} \quad (23)$$

The first line of Eq(23) contains the Galerkin approximation (interior and boundary) and the second line contains the least-squares stabilization. SUPG (Streamline Upwind Petrov Galerkin, see [15] for details) stabilization is obtained by replacing \mathcal{L}^T by $\mathcal{L}_{\text{adv}}^T$. The stabilization matrix $\boldsymbol{\tau}$ is an important ingredient in these methods and is well documented in Shakib [16] and in Franca and Frey [17]. Note that we have chosen to find \mathbf{Y} instead of \mathbf{U} . As

discussed in Hauke and Hughes [18], \mathbf{U} is often not the best choice of solution variables, particularly when the flow is nearly incompressible. For the calculations performed herein, the SUPG stabilized method was applied with linearly interpolated pressure-primitive variables,

$$\mathbf{Y} = \left\{ \begin{array}{c} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{array} \right\} = \left\{ \begin{array}{c} p \\ u_1 \\ u_2 \\ u_3 \\ T \end{array} \right\} \quad (24)$$

By inspecting Eqs(15)-(18) it is clear that all quantities appearing in Eq(23) may be easily calculated from Eq(24).

To develop a numerical method, the weight functions (\mathbf{W}), the solution variable (\mathbf{Y}), and it's time derivative ($\mathbf{Y}_{,t}$) are expanded in terms of basis functions (typically piecewise polynomials; all calculations described here in were performed with a linear basis function. The extension of the quadratic and cubic basis can as well be done on the validated quadratic and cubic bases implementation by[19]). The integrals in Eq(23) are then evaluated using a Gauss quadrature resulting in a system of non-linear ordinary differential equations which can be written as,

$$\mathbf{M}\dot{\underline{\mathbf{Y}}} = \mathbf{N}(\underline{\mathbf{Y}}) \quad (25)$$

where the under bar is added to make clear that $\underline{\mathbf{Y}}$ is the vector of solution values at discrete points (spatially interpolated with the finite element basis functions) and $\dot{\underline{\mathbf{Y}}}$ are the time-derivative values at the same points. Finally this system of non-linear ordinary differential equations is discretized in time via a generalized- α time integrator (see [20]) resulting in a non-linear system of algebraic equations. This system is, in turn, linearized and with Newton's method which yields a linear algebraic system of equations to be solved at each Newton iteration. Newton iterations continue until the non-linear residual is satisfied at each time step, after which the method proceeds to the next time step, starting the process over again.

4 The Discontinuity Capturing Operator

Despite the success of the SUPG, it is well known to be insufficient for flows which contain discontinuities. The SUPG does not preclude overshooting and under-shooting about sharp layers. Therefore, Hughes *et al.* [21] suggested a simple technique for improving resolution of sharp layers while maintaining the optimal rate of convergence. The new method adds an additional “discontinuity capturing term” (DC) which has a form similar to the streamline term, but acts in the direction of the solution gradient rather than in the direction of stream line. The dependence of this term on the solution gradient results in a discrete method, which is nonlinear even when the original PDE was linear. This is of little consequence to the problems that are considered here, since the flow non-linearity is more severe than the new non-linearity introduced by this operator.

The discontinuity operator adds an additional term to the original variational formulation (23) as documented in Shakib [16]. This term provides additional control over gradients in the discrete solution and considerably increases the robustness of the methodology. The DC operator needs to satisfy only a few conditions: (i) in order to control the oscillations, this operator should act in the direction of the gradient, (ii) for consistency it should be proportional to the residual, (iii) for accuracy, it should quickly vanish in smooth regions of the solution. The term derived by Hughes *et al.* is of the form,

$$\sum_{e=1}^{n_{el}} \int_{\Omega^e} \nu^h \hat{\nabla}_{\xi} W^h \cdot [\tilde{\mathbf{A}}_0] \hat{\nabla}_{\xi} V^h d\Omega \quad (26)$$

here $\hat{\nabla}_{\xi}$ is the generalized local-coordinate gradient operator, and ν^h is scalar discontinuity capturing factor of dimension one over time, $[\tilde{\mathbf{A}}_0]$ is the conversion matrix from conservative variables to the entropy variables and \mathbf{V} indicates the entropy variables [16]. The DC operator can be written more explicitly (for our choice of variables \mathbf{Y}) as,

$$\sum_{e=1}^{n_{el}} \int_{\Omega^e} \nu^h g^{ij} W_{,i}^h \cdot [\tilde{\mathbf{A}}_0] \mathbf{Y}_{,j} d\Omega \quad (27)$$

where \mathbf{Y} is the pressure primitive variables described in equation (24) and g^{ij} is the inverse

of the Jacobian of the mapping. The discontinuity factor ν^h is defined as,

$$\nu^h = \max \left\{ 0, \left[\frac{(\mathcal{L}\mathbf{Y} - \mathcal{S}) \cdot \tilde{\mathbf{A}}_0^{-1} (\mathcal{L}\mathbf{Y} - \mathcal{S})}{g^{ij} \mathbf{Y}_{,i} \cdot \mathbf{A}_0^{DC} \mathbf{Y}_{,j}} \right]^{\frac{1}{2}} - \left[\frac{(\mathcal{L}\mathbf{Y} - \mathcal{S}) \cdot \tilde{\tau} (\mathcal{L}\mathbf{Y} - \mathcal{S})}{g^{ij} \mathbf{Y}_{,i} \cdot \mathbf{A}_0^{DC} \mathbf{Y}_{,j}} \right] \right\} \quad (28)$$

5 Finite Element Formulation of the Level Set Method

5.1 Finite element discretization

Equations (3) and (8) can be represented by a single scalar advection equation of the form

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = S \quad (29)$$

In case of the level set equation the forcing function, S , is zero and in the case of re-distancing equation S is given by Eq(9). This section presents the general finite element formulation for the scalar advection equation (29). The solution strategy for solving this equation is similar to flow equations discussed in Section 3.

Again the spatial discretization is performed using the finite element method, and the finite element approximation spaces namely the solution and the weight function space are as defined in Section 3. In keeping with the approach of finite element method described above write Eq(3) in the residual form. To derive the weak form of residual form, Eq(3) is multiplied by a smooth weighting function w belonging to a space of functions $w \in \mathcal{W}_h$ (the scalar counterpart of the vector space defined in Section 3). The product is then integrated over a spatial domain. As the equation is solved in the convective form, we do not integrate by parts thus there are no boundary integrals. The resulting weak form is: find $\phi \in \mathcal{V}_h$ such that,

$$\int_{\Omega} (w \phi_{,t} + w u_i \phi_{,i} + w S) d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \mathcal{L}^T w \tau (\phi_{,t} + u_i \phi_{,i} - S) d\Omega = 0 \quad (30)$$

Where τ is the stabilization parameter (defined earlier). By replacing \mathcal{L}^T by \mathcal{L}_{adv}^T (for the scalar equation $\mathcal{L}_{adv}^T = u_i \frac{\partial}{\partial x_i}$), we obtain the SUPG (Streamline Upwind Petrov Galerkin, see [15] for details) stabilization. Next, the weight functions (w), the solution variable (ϕ),

and its time derivative (ϕ_t) are expanded in terms of linear basis functions as before. The integrals are then evaluated using Gauss quadrature resulting in a system of non-linear ordinary differential equations which can be written as

$$\mathbf{M}\dot{\phi} = \mathbf{N}(\phi) \quad (31)$$

The solution of this system is analogous to that of the flow. The two systems (flow and scalar) are iterated until convergence is achieved.

5.2 Volume constraint

During the iteration of vectors and scalar the interface is convected with the local flow speed, which, as mentioned before destroys the distance function. As mentioned earlier, this distance function is restored by the solution of Eq(10) via a finite element discretization as described above. During this re-distancing step, additional care is taken to restrict the interface from moving. To constrain the interface, we implemented the strategy proposed by Sussman *et al.* [5] for the finite difference method. The principle behind the constraint calculations is to enforce the volume filled by each phase in an element to remain constant, when the re-distance step is applied. In order to minimize the volume variation, we project the current values of the level set function, denoted as \tilde{d}^k , onto new values, denoted by d^k which satisfy,

$$\int_{\Omega^e} H'_\epsilon(d^0)(d^k - d^0)d\mathbf{x} = 0. \quad (32)$$

where,

$$H'_\epsilon(d) = \left\{ \begin{array}{ll} 0, & \text{if } |d| > \epsilon, \\ \frac{1}{2} \left[\frac{1}{\epsilon} + \frac{1}{\epsilon} \cos\left(\frac{\pi d}{\epsilon}\right) \right], & \text{if } |d| \leq \epsilon. \end{array} \right\} \quad (33)$$

The projection described in equation (32) is implemented by assuming d^k has the form,

$$d^k = \tilde{d}^k + \lambda_{\Omega^e}(\tau^k - \tau^0)H'_\epsilon(d^0) \quad (34)$$

where λ_{Ω^e} is assumed to be constant in Ω^e , and is given by,

$$\lambda_{\Omega^e} = \frac{-\int_{\Omega^e} H'_\epsilon(d^0) \left(\frac{\tilde{d}^k - d^0}{\tau^k - \tau^0} \right) d\Omega^e}{\int_{\Omega^e} (H'_\epsilon(d^0))^2 d\Omega^e}. \quad (35)$$

In the current algorithm the integrals for estimating λ in (35) are evaluated at the element level and are projected onto the global nodes by L^2 projection. Then the equation (34) is solved to obtain the constrained re-distanced level set function. This step is applied after each re-distancing of the level set field.

6 Modeling of Surface Tension

In the present work, fluid interfacial motion induced by surface tension may play a significant role. The surface tension force is a result of the uneven molecular forces of attraction experienced by fluid molecules near the interface. Surface tension creates a microscopic, localized surface force that exerts itself in both tangential and normal directions. We use the continuum approach proposed by Brackbill *et al.* [10] in order to represent the surface tension force as a body force. This model represents surface tension as a continuous three dimensional effect across an interface, rather than as a boundary value condition at the interface. The resulting body force due to surface tension can be written as

$$\mathbf{F} = -\frac{\kappa(\phi)\nabla H(\phi)}{W} \quad (36)$$

where $\kappa(\phi)$ is the curvature defined as

$$\kappa(\phi) = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \quad (37)$$

and $H(\phi)$ is the Heaviside function as defined in equation (7). W is the Weber number given by

$$W = \frac{\rho_1 L U^2}{\sigma} \quad (38)$$

The surface tension term and local inter-facial curvature are easily represented in terms of the level set function. As the level set function in our formulations is a signed distance from the interface, the curvature can be accurately computed from the level set function. The form of the surface tension force as body force used here is due to Chang *et al.* [22].

7 Ghost Fluid Method

Although Eulerian schemes work well for most compressible flows, they have been shown to admit spurious non-physical oscillations near the material interfaces. Fedkiw and Aslam, *et al.* [23] proposed a new numerical method for treating interfaces in Eulerian schemes that maintain a Heaviside profile of the density with no numerical smearing. They also used the level set function to track the interface in their work, in addition they used ghost cells (actually ghost nodes in their finite difference frame work) to prevent the smearing of the density across the interface. The motivation for their method stems from the fact that the non-physical oscillations encountered by Eulerian schemes across the material interface are due to the radical change in the equations of state across the material surface. On the other hand Lagrangian schemes won't smear the density profile, and it is clear which equation of state is valid at each location. But these Lagrangian schemes do have their own problems when subjected to large deformations. A good summary of both Eulerian and Lagrangian schemes is presented in [24]. The original method proposed by the authors tracks the interface with a level set function which gives the exact sub-cell interface location. At this interface, they solve an approximate Riemann problem similar to the methods in [25], and [26].

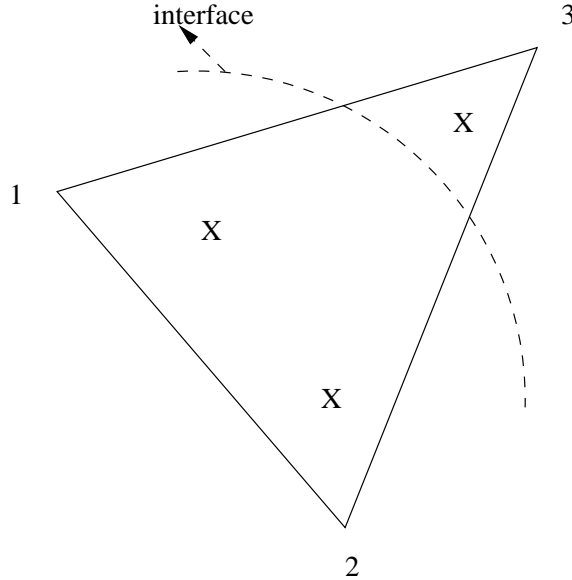


Figure 1: A typical triangular element containing interface

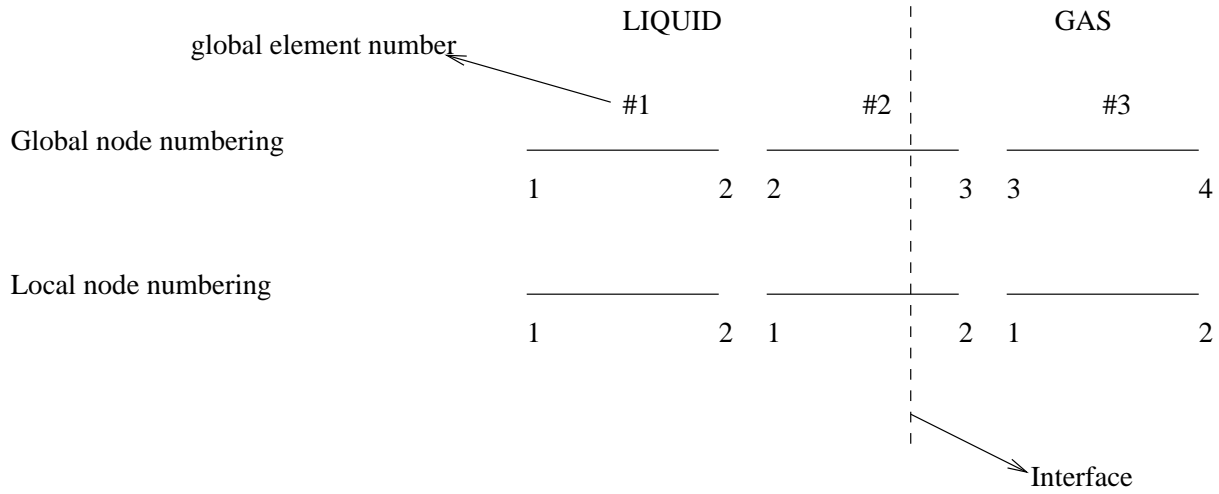


Figure 2: A 1-D example of global and local numbering

The approach taken in our work draws heavily on the method of Fedkiw and Aslam, *et al.* [23], appropriately modified for a finite element frame work. The description of the method is as follows. In our method we also use a level set function to keep track of the interface and the zero level set represents the location of the interface. The positive level sets representing the heavier fluid and the negative values representing the lighter fluid. Each fluid satisfies a different equation of state on each side. The clear demarcation can be made from the exact representation of the interface location by the zeroth level set. Then we define a ghost element in addition to the existing element for every element which is intersected by the interface. Hence the elements which contain the interface have the mass, momentum and energy of both the fluids. This is done by evaluating the integrals in Eq(23) accordingly. Figure 1 shows a typical two dimensional triangular element which is intersected by an interface. For example, assume that the nodes 1 and 2 are in the liquid phase, while node 3 lies inside the gas phase. So for this element the integrals over the element are evaluated assuming once the whole element is occupied by the liquid phase and next by assuming it is entirely occupied by the gas phase. While building the nodal contributions to both the right hand side and the left hand side, a choice is made between the liquid or gas integral depending on whether the node lies in liquid or gas phase. So in the above example while evaluating the local residual at node 1, we choose the liquid integral, on the other hand the local residual at node 3 is evaluated based on the integral evaluation for gas phase.

The method can be further explained by the following simple 1-D example. Consider the simple 1-D domain shown in Figure 2. The domain is divided into three elements, and the second element is intersected by the interface. Let us assume that the region to the left of the interface contains the liquid phase and the region to the right contains the gas phase. The figure shows both the local and the global node numbering. The finite element discretization leads to the ordinary differential equation,

$$\sum_{B=1}^{nel_{node}} \mathbf{W}_B \mathbf{G}_B = 0 \quad (39)$$

Which implies if the weight functions are chosen to be arbitrary, then $\mathbf{G}_B = 0$, where \mathbf{G}_B is obtained by assembling the local \mathbf{G}_b^e . Note the 'B' indicates the global node numbering, the subscript 'b' indicates the local node number and the super script 'e' indicates the element number. For the above 1-D example the global assembly operation will be defined as follows,

$$\mathbf{G}_1 = \mathbf{G}_1^1 \quad (40)$$

$$\mathbf{G}_2 = \mathbf{G}_2^1 + \mathbf{G}_1^2 \quad (41)$$

$$\mathbf{G}_3 = \mathbf{G}_2^3 + \mathbf{G}_1^3 \quad (42)$$

$$(43)$$

The element level residual \mathbf{G}_b^e can be defined as

$$\mathbf{G}_b^e = \int_{\Omega^e} g_b^e(\tilde{\xi}) d\Omega^e = \sum_{k=1}^{nqpt} g_b^e(\tilde{\xi}_k) \cdot \mathbf{W}_k \quad (44)$$

where,

$$g_b^e(\tilde{\xi}_k) = \sum_{b=1}^{nenl} g^e(\tilde{\xi}_k) \cdot N_b \quad (45)$$

In the above equations 'nqpt' denotes the number of Gaussian quadrature points and the 'nenl' indicates the number of element nodes. The ghost fluid method calculates $g^e(\tilde{\xi}_k)$ twice for the elements that contain an interface. Once assuming the whole element is occupied by the liquid and once by assuming this occupied by the gas. While evaluating $g_b^e(\tilde{\xi}_k)$, depends upon whether node b contains the liquid phase or gas phase, $g^e(\tilde{\xi}_k)$ is selected. In essence the method assumes a ghost fluid in all the elements that contain interface. This ghost fluid behaves as it is a pure liquid for the nodes that are in the liquid phase and behaves as pure gas when seen from the nodes that are in gas phase.

8 Adaptive Mesh Strategy

Adaptive strategies are an important tool in efficient finite element computations. Adaptivity affords the opportunity to obtain numerical solutions with a controlled accuracy with minimum degrees of freedom. In addition adaptive refinement procedures have become important for increasing the reliability and reducing the cost of numerical computations in many engineering problems. The current research deals with the problems that involve multiple shocks with varying strength, regions with discontinuous solutions, and rapidly deforming interfaces. For such kind of problems the error in discretization may result into a poor prediction of the actual physics. Hence it is important to maintain a high resolution of the mesh in the regions that have large solution gradients. Also it is well known that the smoother regions of the flow often benefit from p -refinement, while regions with discontinuities or sharp layers are better resolved using h -refinement. Despite the increase in the computational resources, uniformly refined meshes for large scale three dimensional problems will lead to an inefficient, and often, impractical computations. Hence, in the present work, an adaptive strategy was implemented to refine and coarsen the mesh in regions of steep gradients. The essential ingredients of an adaptive procedure are:

1. A tool for assessing the error of the solution computed with a given mesh.
2. An algorithm to define a new spatial discretization via h and/or p -refinement.

8.1 Error indicators implemented

Two different approaches can be used for assessing the error: error estimators and error indicators. Error estimators approximate a measure of the actual error in a given norm. Error indicators, on the other hand, are based on more heuristic considerations. Hence an error indicator only gives relative information about the error, but is typically easy to compute and implement in the analysis code. In fluid mechanics computations, which involve non-self-adjoint operators, little progress has been made on error estimators. For this reason most of the studies in the fluid dynamics rely on local error indicators to refine the mesh

without specifying a total error. In PHASTA-2C, error indicators were utilized for the adaptive procedure.

There are many error indicators available in the literature. Most of them are similar and based on the second gradient of a key variable. In the present studies a total of ten error indicators are generated at all the nodes in the computational domain. The first three error indicators are based on the L_2 norm of the residual of the three components of the momentum equation. The next three are L_2 norm of the normal component of the shear stress. The last four are based on the difference in the velocity and pressure at the current time step to the average of the of the solutions obtained from the two previous time steps. Although there is no general theory to guide the use of the last four error indicators, the preliminary results seem promising. The ten error indicators used in the present work:

$$e_i = \int_{\Omega^e} N_a(\mathcal{L}u_i - f_i)^2 d\Omega \quad i = 1, 2, 3 \quad (46)$$

$$e_i = \int_{\Omega^e} N_a(\tau_{ij,j})^2 d\Omega \quad i = 4, 5, 6 \quad (47)$$

$$e_i = \sum_{n=1}^{nstep} (u_i - \bar{u}_i)^2 \quad i = 7, 8, 9 \quad (48)$$

$$e_i = \sum_{n=1}^{nstep} (p - \bar{p})^2 \quad i = 10 \quad (49)$$

Where \bar{u}_i and \bar{p} are the average velocity and pressure interpolated between the solution values at n th and $n + 1$ time steps. When computing two-phase flows an additional error indicator must be considered. The above indicators are only weakly linked to the quality of the representation of the the interface. Though the level set field is smooth, it does have a discontinuous slope at the interface. The second derivative of the scalar field is therefore a viable choice for indicating the location in need of refinement/high-resolution. This can be readily achieved with the already available information in the algorithm. Our aim is not only to refine, but refine it smooth enough around the interface. For this purpose we have used the level set Heaviside function property to smooth the refinement at and near the interface.

8.2 Refinement strategy

In PHASTA-2C the mesh generation tools developed at the Scientific Computation Research Center (SCOREC) were extensively used. The mesh enrichment is done such that the individual elements are subdivided without altering their original location. As a first step the model and the original mesh are loaded in to the adaptive code. Then the parameters to control the adaptive procedure given through an input file and the values of the error indicators at each node are read in. Based on this information the quantitative parameters for re-meshing are chosen. Then the edges are marked based on the error measure. The marking is allowed to propagate for a specified (typically on or 2) number of layers. This serves two purposes. First it decreases the frequency that adaptivity must be carried out. Second, it allows the mesh to change size in a smoother manner which is known to reduce error.. Once the edges are marked, “mesh adapt” (a SCOREC tool) is used to refine the mesh, and once the refinement is done, the mesh is smoothed.

8.2.1 Edge marking

The decision as to whether a mesh edge is to be marked or not is made based on both error value and model classification information. Only those mesh entities whose errors are greater than a threshold value and are classified on the closure of a prescribed entities are marked. Three types of marking strategies were adopted.

1. mark a mesh vertex, if it's error is greater than the threshold
2. Mark a mesh edge if error at it's either of the bounding vertices is greater than than the threshold value.
3. Mark a mesh face if error at it's any of the bounding vertices is greater than the threshold value.

8.2.2 Mesh quality

Maintaining the mesh quality is one of the vital factors for a successful adaptive procedure. The mesh quality was preserved by the following given conventions:

1. Mark the third edge if two of the edges of a face are marked
2. Always mark the longest edge if any of a mesh face is marked
3. If the original model has matched faces, always keep the marked edges on those faces consistent.

9 Numerical Results and Discussion

The goal of this research is to simulate single and multiple bubble dynamics, however it would be fool hardy to apply the new method described above into such a complex problem without first validating that all of the relevant physics can be predicted on more simple, well understood test problems. This was the motivation for the incompressible multi-phase flow simulations presented in Section 9.1. Through these simulations the ability of the level set method to represent the interface and robustness of the method for the topological changes has been tested and compared to the available results in the literature. Section 9.2 focuses on the compressible flow simulations. In particular, in this section we will demonstrate that the newly developed code can: *(i)* capture shocks correctly, *(ii)* represents the compressible gas dynamics accurately, *(iii)* tracks very complex bubble motions including the bubble implosion very efficiently, *(iv)* make use of unstructured and adapting grids.

9.1 Simulation of Incompressible multiphase flows

9.1.1 Simple advection of a cylindrical bubble

To assess the ability of the algorithm, a simple test problem of advecting a cylindrical bubble through a rectangular domain was considered. The liquid is flowing with a constant velocity of 1m/sec in vertical direction. The buoyancy force is not activated. Hence the bubble

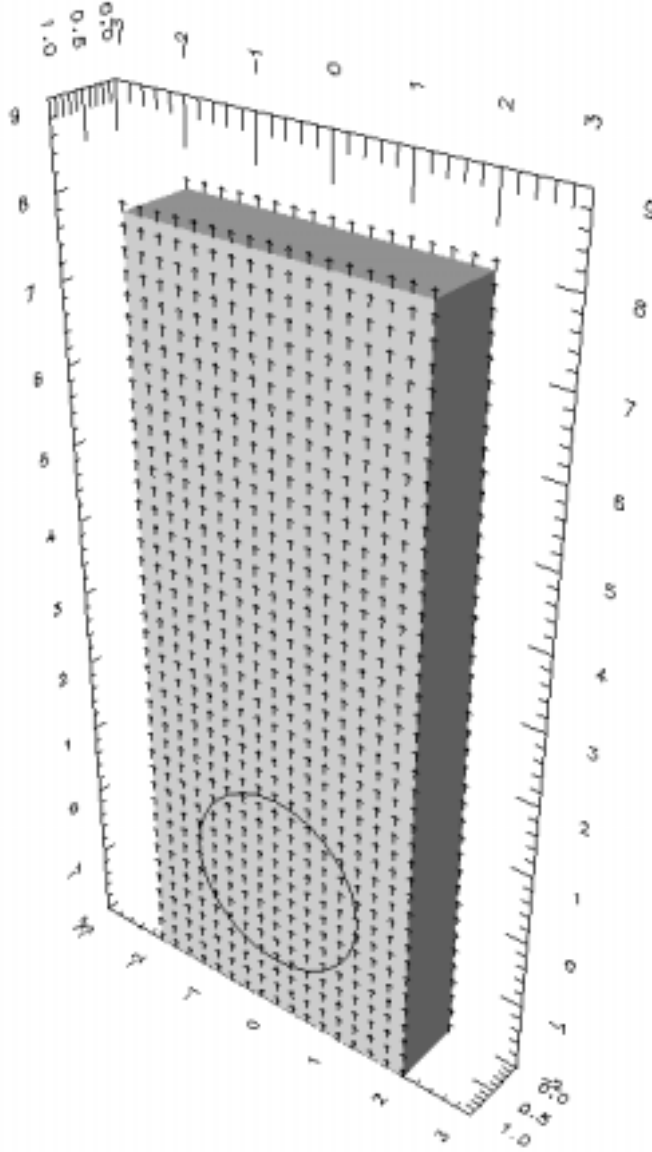


Figure 3: The computational domain and bubble position at $t = 0$.

should simply advect through the domain with the velocity of the liquid. The domain and the initial position of the bubble are shown in the Figure 3. The computational domain dimensions are $-2.0 \leq x \leq 2.0$ $-2.0 \leq y \leq 8.0$, and $0 \leq z \leq 1$ and the mesh consists of

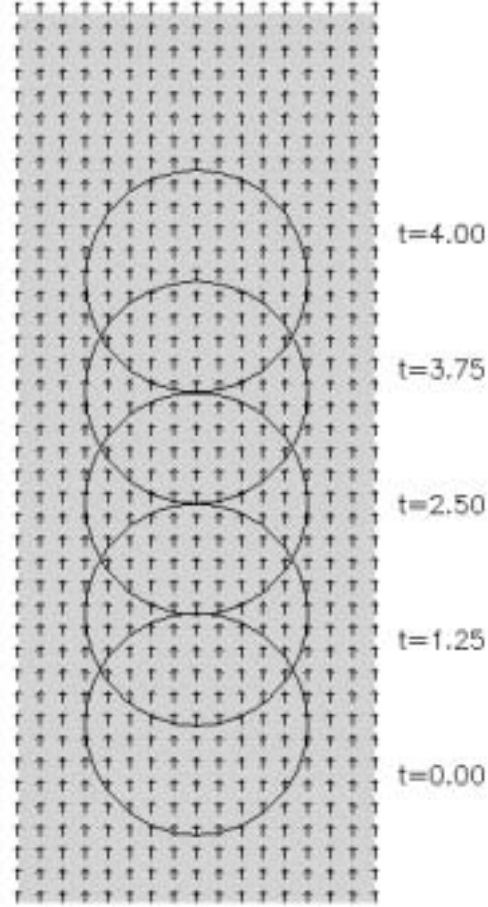


Figure 4: Bubble position at various time instants

640 hexahedral elements ($16 \times 40 \times 1$). Periodic boundary condition is applied in x- and z- directions. An uniform velocity of 1m/sec is specified at the inlet and a constant pressure boundary condition is applied at the exit of the domain. The bubble radius is 1.25m, and the

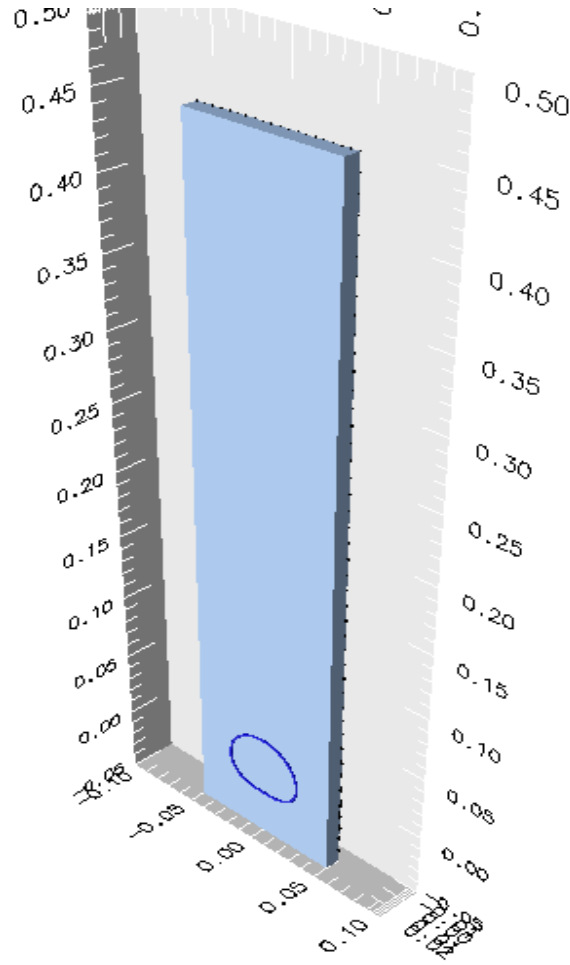


Figure 5: The bubble position at $t = 0$

bubble location is such that the bottom of the bubble is positioned at -1.25m. The motion of the bubble through the domain is shown in the Figure 4. Clearly the bubble is advected with the fluid velocity of 1m/sec and also the shape of the bubble is preserved along with the conservation of the volume of the bubble.

9.1.2 Two dimensional Simulation of bubble motion under buoyancy force at a medium Reynolds Number

The rise and the deformation of a 2-D gas bubble in an otherwise stationary liquid contained in a vertical, rectangular container is investigated using the level set method incorporating surface tension stresses.

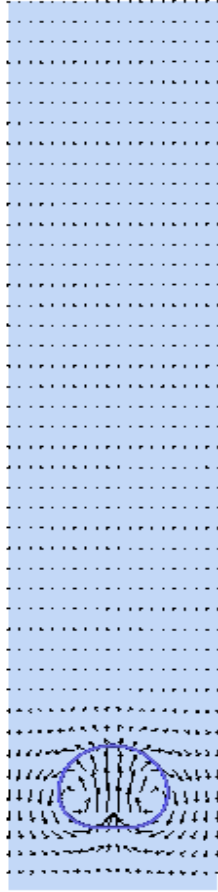


Figure 6: Evolution of rising bubble under buoyancy at $t = 0.05$

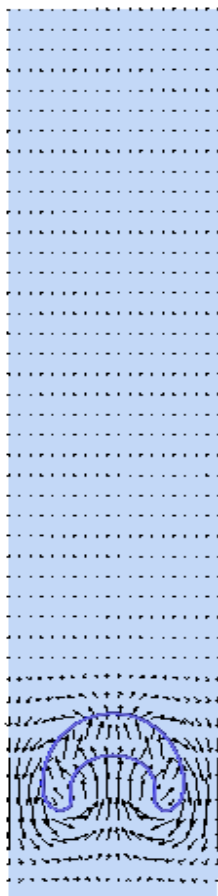


Figure 7: Evolution of rising bubble under buoyancy at $t = 0.15$

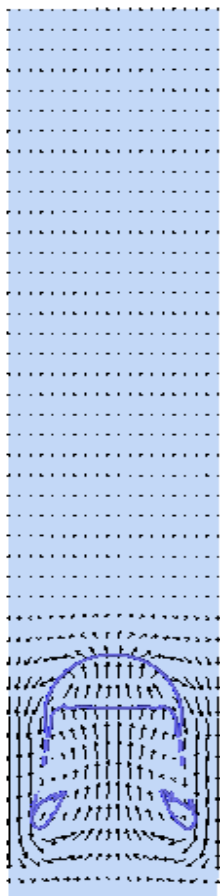


Figure 8: Evolution of rising bubble under buoyancy at $t = 0.3$

The density of the water and air are taken to be, $1000 \frac{kg}{m^3}$, and $1.226 \frac{kg}{m^3}$, respectively, which gives a density ratio of 1,000. The viscosity of liquid phase is taken as $3.5e^{-1} \frac{kg}{m.sec}$, and the viscosity of the air to be, $3.58e^{-3} \frac{kg}{m.sec}$, so that the viscosity ratio between the two phases is 100. The domain width is 0.1m and the domain length is 0.5m. The initial and the boundary conditions are shown in the Figure 5. The initial radius of the bubble is 2.5cm. With the above parameters, if we define the Reynolds number as $Re = \frac{(2R_b)^{3/2} \sqrt{g} \rho_c}{\mu_c}$, and Bond Number as $B = \frac{4\rho_c g R^2}{\sigma}$ then $Re = 100$, and $B = 200$. Grace *et al.* [27] presented a diagram showing the effect of fluid properties and the equivalent bubble diameter on the shape and the terminal velocity of an isolated bubble. According to their experiments for the parameters chosen above ($Eo=200$, $M=0.08$), formation of a skirted bubble is reported. The mesh used in simulations consists of $100 \times 500 \times 1$ elements. Periodic boundary condition is imposed in z direction. On the side planes, the normal component of the velocity (x-directional velocity) is specified as zero, and a zero traction is also imposed. A constant pressure boundary condition is prescribed at the outflow, and a zero velocity is assigned on the bottom of the container.

When the bubble begins to rise owing to the buoyancy force acting on the bubble, the pressure gradient at the lower surface of the bubble is higher than on the top surface, and the vortex sheet which develops at the surface has a rotation (Figure 6) which induces the motion of a jet of water that pushes into bubble from below. This phenomena is captured within the numerical simulations presented in Figure 6, and Figure 7. These are the solutions at the time instances of $t = 0.05$ and $t = 0.15$, respectively. At this stage the jet does not effect the liquid above the bubble. The velocity of the upper surface of the bubble in comparison to the rest of the bubble is low resulting into the shell form of bubble. With time, the water jet from the bottom further pushes the bubble, which causes the lower surface to move more towards the upper cap of the bubble (Figure 8) forming a skirted bubble. Eventually the liquid jet pinches off the bubble and shreds satellite bubbles as shown in the Figure 8. This result is consistent with the findings of Delnoij *et al.* [28] who also observed the formation of spherical cap bubble and shredding of satellite bubbles. Walter *et al.* [29] also observed the detachment of two small bubbles at the lower extremities of the main bubble, during the rise of their two dimensional bubble. The solutions obtained from this method are also in good

agreement with the Sussman *et al.* [5], who also carried out the simulation for the same Reynolds and Bond number. In the simulations presented here actual piercing of the bubble does not occur (leading to a toroidal bubble) due to the relatively low Reynolds number. For a low Reynolds number, the liquid jet below the bubble is not strong enough to pierce causing the bubble instead to rise as a cap as seen in Figure 8. The level sets remain a distance function despite the changing bubble topology. The advantage of the levels sets is also demonstrated as the shredded bubbles motion was predicted without altering the algorithm.

9.1.3 Simulation of a three dimensional spherical bubble motion under buoyancy force

To further demonstrate the capability of PHASTA-2C, a three-dimensional simulation of the motion of the bubble was performed. Starting from a perfectly spherical bubble which is initially at rest, the buoyant motion of the bubble is studied by tracking the interface using the level set method.

For the full three dimensional simulations a $32 \times 80 \times 32$ elements mesh was adopted. The initial position of the bubble is shown in the Figure 9. The initial and boundary conditions are same as the simulations presented in Section 9.1.2. The ratio of the rectangular column width to the bubble diameter is 1.6 with 20 elements across the bubble diameter. Again a Weber number of 100 is chosen for the simulations presented here.

As described earlier the liquid jet from the bottom tries to push the lower surface of the bubble, which can be seen in Figure 10. The deformation of the bubble from a sphere to an elliptic cap can be seen in Figure 11. Figure 12 shows the further deformation of the bubble due to the impinging jet from the bottom. The view shown in Figure 12 is from bottom, in which we can see the the liquid pushing the bottom surface closer to the top, but not yet piercing the top surface. Eventually, at time $t = 1.25$, the jet penetrates the liquid bubble and lower surface pierces the top surface, which can be seen clearly in the Figure 13. Thereafter the formation of the toroidal bubble is complete. Figure 14 shows the corresponding side view of the bubble at time, $t = 1.25$. Figure 15 indicates that the toroidal

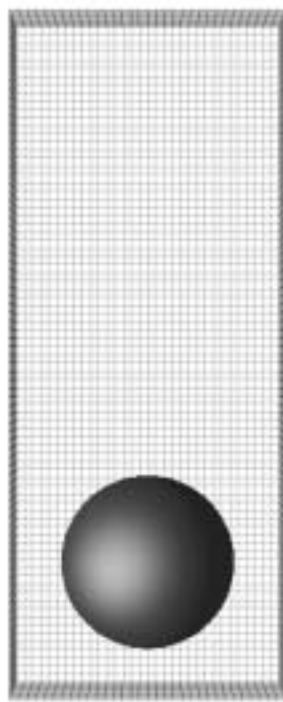


Figure 9: Initial position of the spherical bubble. Here only the zero value of the level set is shown to illustrate the bubble surface.

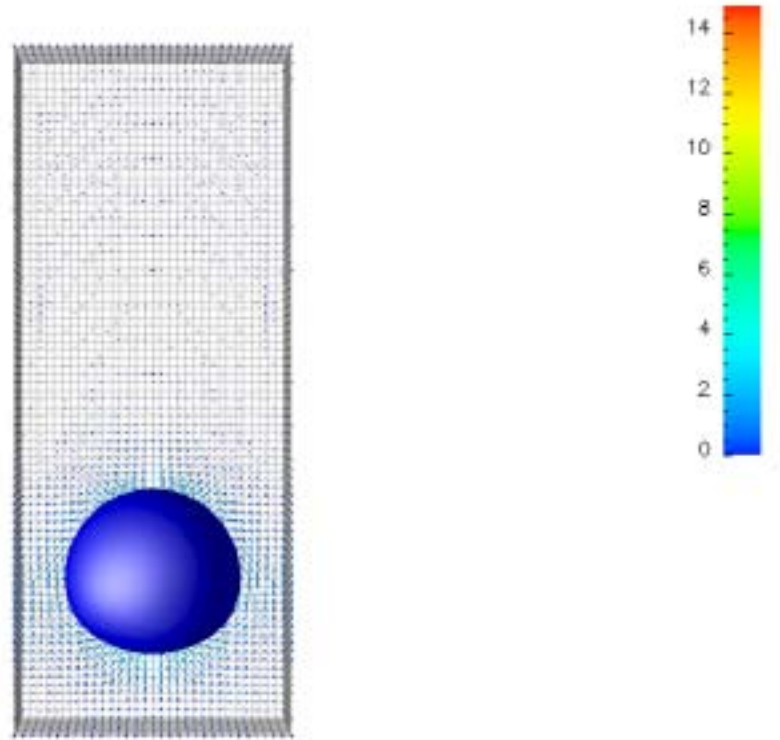


Figure 10: Shape of the spherical bubble at $t = 0.25$. Surface as before with velocity vectors added to show fluid motion.

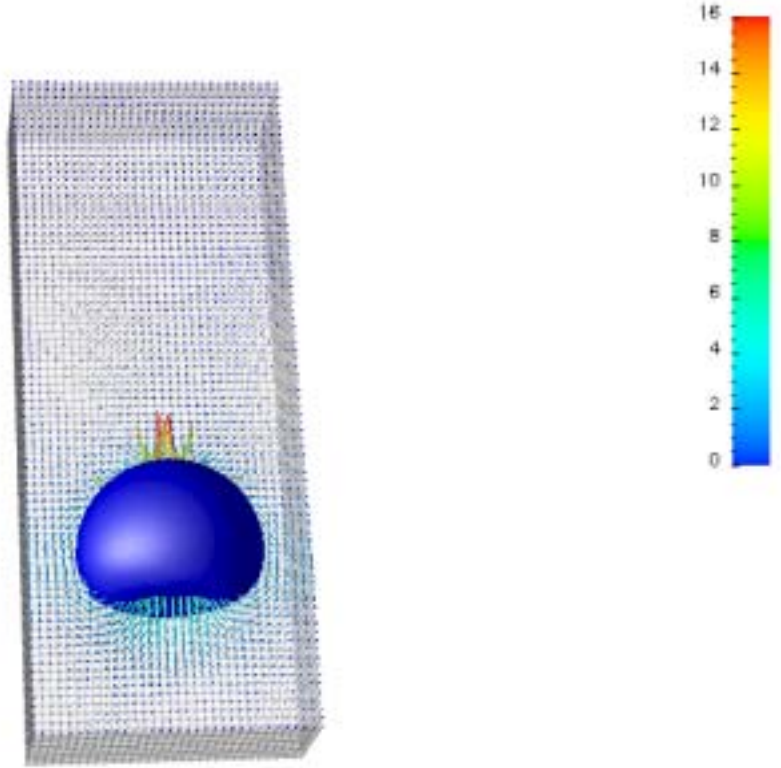


Figure 11: Shape of the spherical bubble at $t = 0.375$.

bubble is about to break up; note the thin air pocket at the top, which is about to be carried away into the liquid above the bubble. The vorticity in the bubble surface is transferred to circulation about this annular toroidal bubble, and ultimately this toroid breaks up into a ring of small bubbles in order to preserve the circulation. The features of three dimensional bubbles reported from the experiments of Walters *et al.* [29] included the deformation of the lower surface of the bubble by a jet of liquid, forming a so called liquid tongue, “the piercing of the top surface and the resultant formation of a toroidal bubble with a very small bubble in the center”, and an increase in the diameter as it rises further. We have also observed an increase in inner and the outer radii of the toroid as it rises, and an expansion of toroidal bubble.

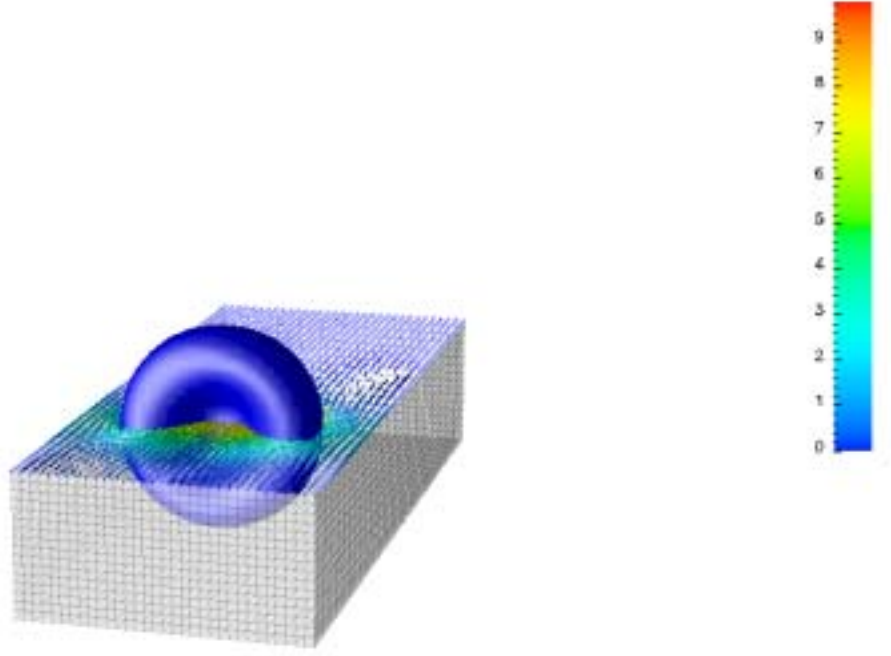


Figure 12: Rising spherical bubble at $t = 0.5$.

9.2 Simulation of Compressible flows

9.2.1 Steady shock problem

To demonstrate the discontinuity capturing operator, and the refinement strategy, a classical standing shock problem was considered. The selected problem consists of two flow regions separated by a shock. If the flow properties on the two sides of the shock satisfy the normal shock conditions, the shock will be stationary. The following initial conditions satisfy these conditions.

$$\left\{ \begin{array}{l} M = 2 \\ \rho = 1 \\ u = 1 \\ p = 0.1785 \end{array} \right\} \text{ for } x < 0 \text{ and } \left\{ \begin{array}{l} M = 0.57735 \\ \rho = 2.66667 \\ u = 0.37500 \\ p = 0.80357 \end{array} \right\} \text{ for } x > 0 \quad (50)$$

($\gamma = 1.4$ and $c_v = 716.5$)

The boundary conditions are set to the above data, and the y- and z-components of the

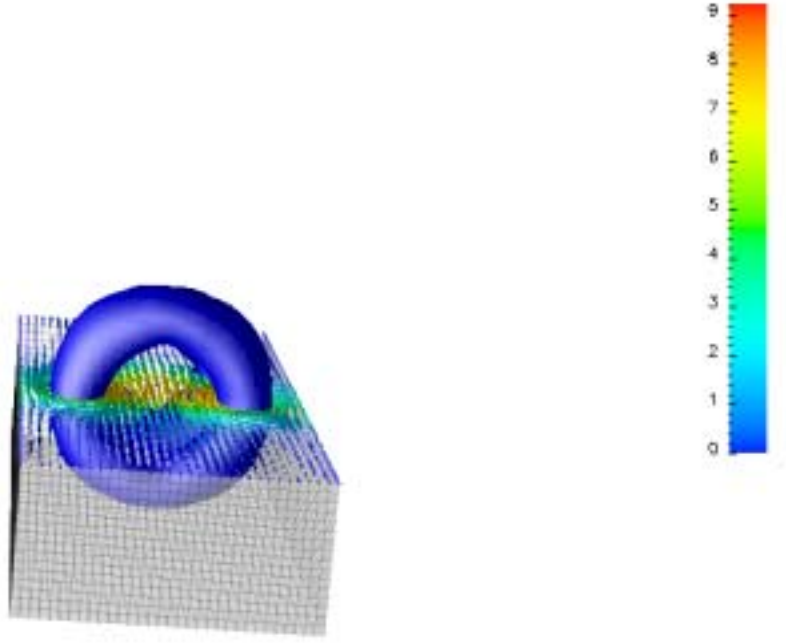


Figure 13: Rising spherical bubble at $t = 1.25$.

velocity are set to zero on the entire domain. The computational domain dimensions are $-19.5 \leq x \leq 19.5$ $-0.5 \leq y \leq 0.5$, and $0 \leq z \leq 1$ and the initial, mesh consisted of 39 hexahedral elements ($39 \times 1 \times 1$).

The computed pressure using the Galerkin/least squares without the discontinuity term is shown in the Figure 16. From the figure it is clear that the original position of the shock is maintained by the method, which indicates that the method is indeed a flux conservative method. But the solution has small undershoots and overshoots near the shock. These are however, very localized and do not corrupt the solution, a small distance away from the shock. However, for the problems we are interested in, it is essential to capture these discontinuities to represent the relevant physics of the problem accurately.

Figure 17 presents the computed pressure using the discontinuity capturing operator. The solution obtained is free of under-shoots and overshoots. This shows that the discontinuity capturing operator in PHASTA-2C is able to control the oscillations as desired. Moreover, one can see from the figures that the steady shock conditions are preserved resulting in a

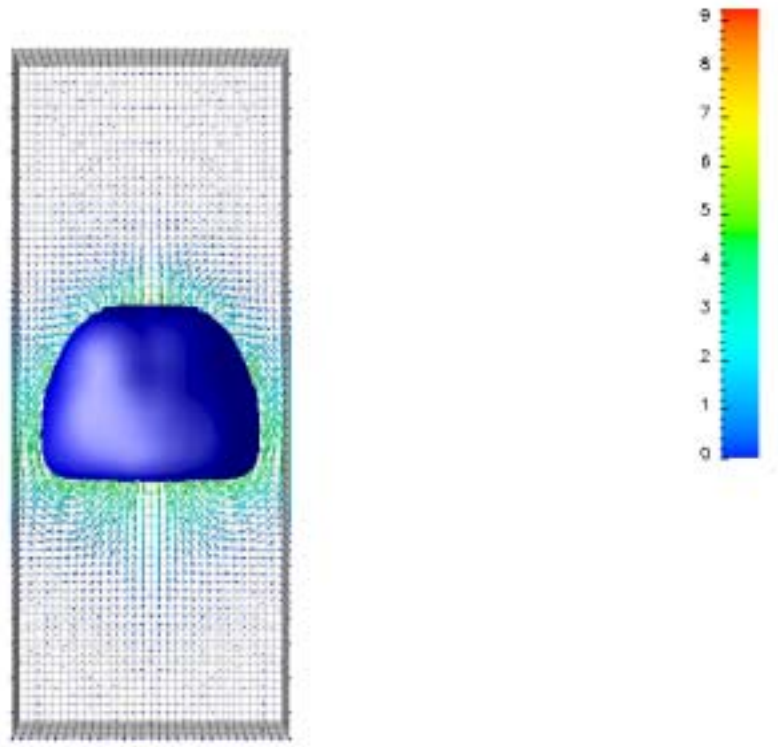


Figure 14: Rising spherical bubble at $t = 1.25$ (front view).

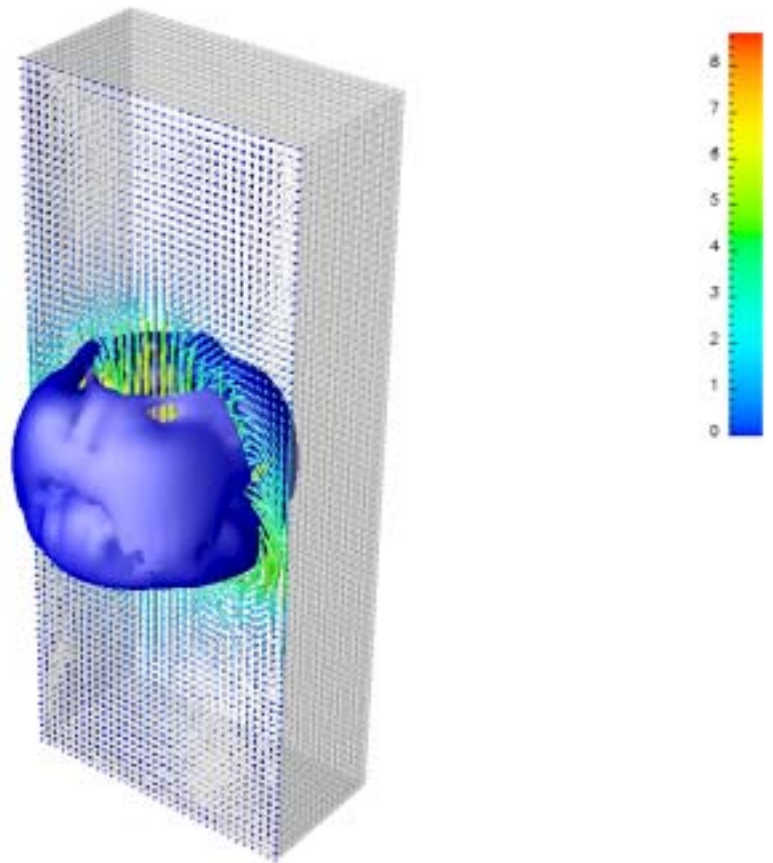


Figure 15: Rising spherical bubble at $t = 1.5$.

non-smeared shock.

9.2.2 Sod's shock tube problem

The purpose of this test is to demonstrate the capability of the code to predict the compressible problems having the sharp discontinuities in the solution. A good test problem to verify the accuracy of the numerical method and the resolution of the shock is a shock tube problem, has been suggested by Sod [30]. The problem consists of a two different material states separated by a diaphragm. Initially the fluid is at rest and $\rho_1 = 1.0, p_1 = 1.0, u_1 = 0.0$ on one side of the shock and $\rho_2 = 0.125, p_2 = 0.1, u_2 = 0.0$ on the other side of the shock. At time $t=0$ the diaphragm is broken, and a shock wave travels to the right and a rarefaction wave moves towards left. Farther down the tube there will be a contact discontinuity. The solution obtained with a resolution of 100 elements at $t=0.2\text{sec}$ is shown in the Figure 18. From the figure it is clear that the method is very well able to resolve all the shocks and

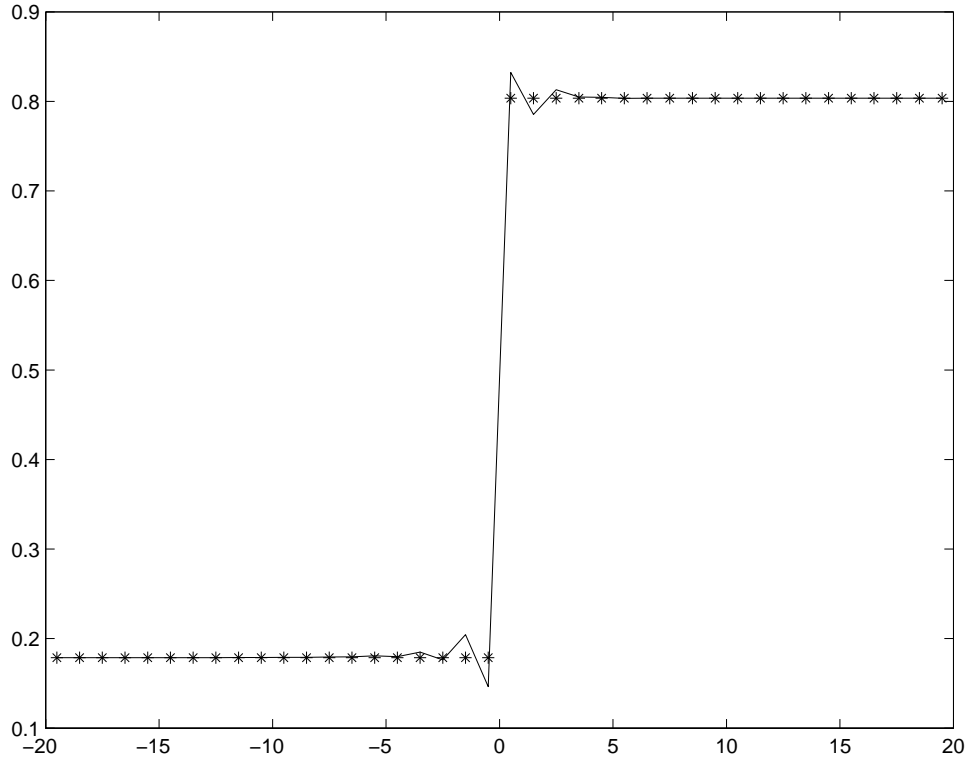


Figure 16: Pressure predictions without the DC operator: ---- * exact solution, — SUPG solution

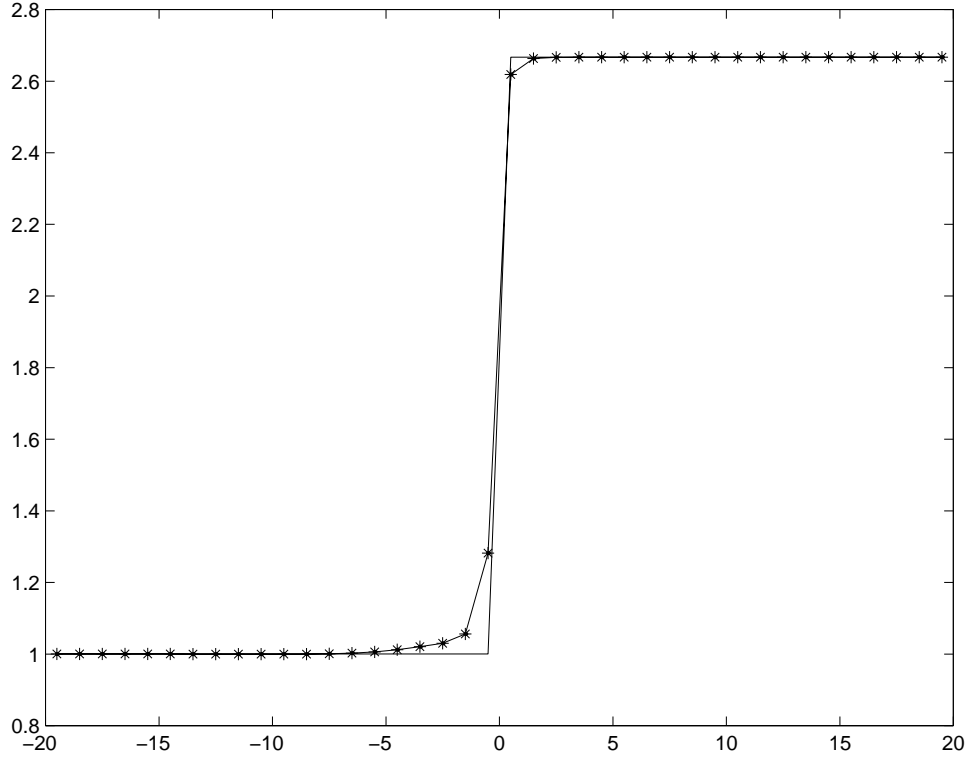


Figure 17: Pressure predictions with the DC operator: — exact solution, ---- * SUPG with DC operator solution

the discontinuities very accurately. Also we can see that, as expected, across the contact discontinuity the pressure and the velocity are continuous although the density and the specific energy are discontinuous. Both the shock wave moving to the right and the rarefaction waves are predicted well and at correct positions. Moreover, the predicted solutions are free of numerical oscillations. The solution obtained compare very well with the numerical results obtained using high resolution schemes, and modern upwind schemes.

9.2.3 Implosion of a spherical shock wave using an adaptive mesh strategy

Next we considered the simulation of a strong spherical shock wave traveling to the center of a spherical domain. Initially the spherical domain consists of a gas of uniform density and pressure. A strong shock wave is generated at $t=0$. The wave could have been generated, for example, by a “spherical piston” which pushed the gas inward, imparting to it certain amount of energy. The origin of the shock wave does not affect the solution, except the

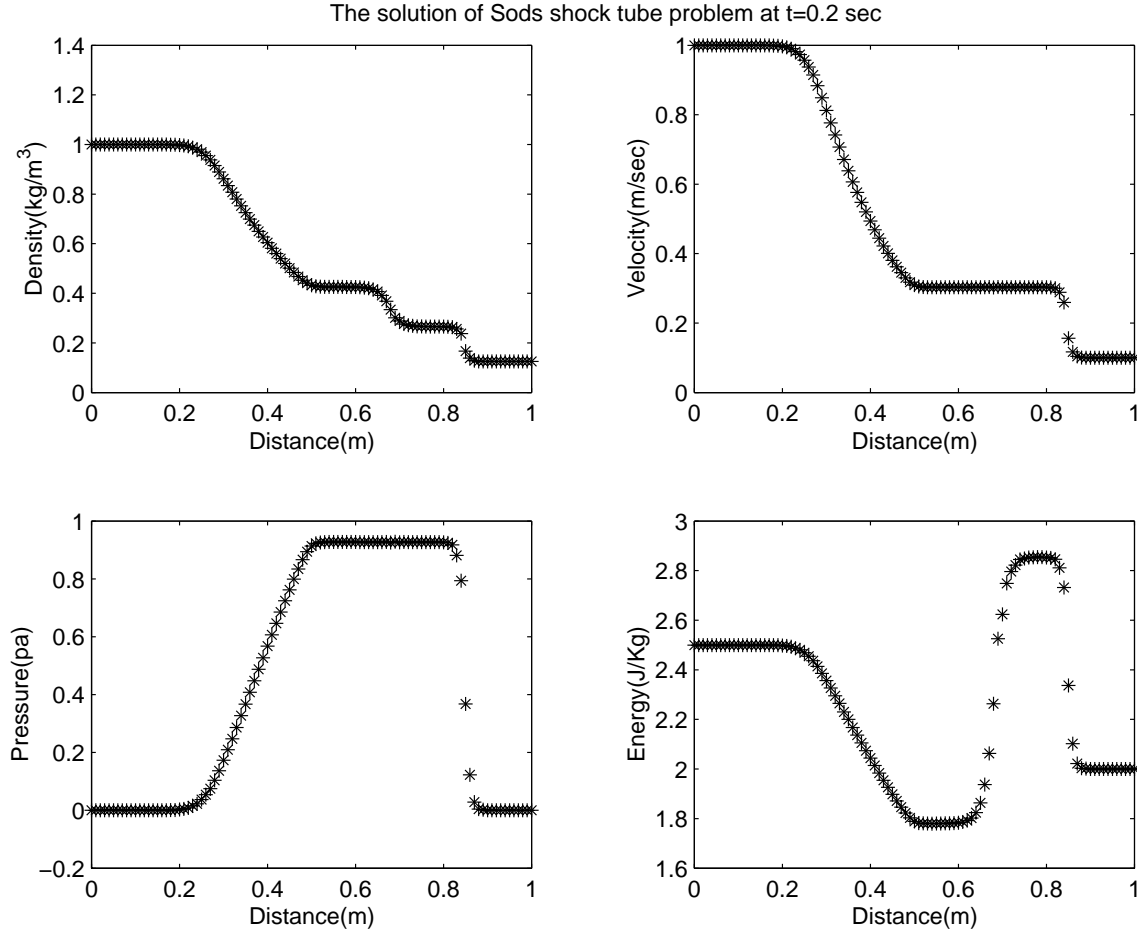


Figure 18: Sod's shock tube problem

wave generated is spherically symmetric. As the wave converges to the center the energy becomes concentrated at the front, and the wave gets strengthened. At the instant of shock front reaches the center of the spherical domain, the problem becomes nearly singular. This exercise clearly demonstrates the code's capability to handle the intensifying shocks. To resolve the strong shocks we need a really refined mesh. As the problem is fully three dimensional, if we do the uniform refinement the mesh would be enormous. As the regions away from the shock front does not need the fine resolution we need refine mesh only around the shock front using the adaptive refinement strategy. As the shock moves in we coarsen the regions which were refined earlier and refine the regions where the new shock front is located. This strategy results in excellent savings in the computational cost.

The initial conditions are shown in the Figure 19. These conditions generate a shock wave

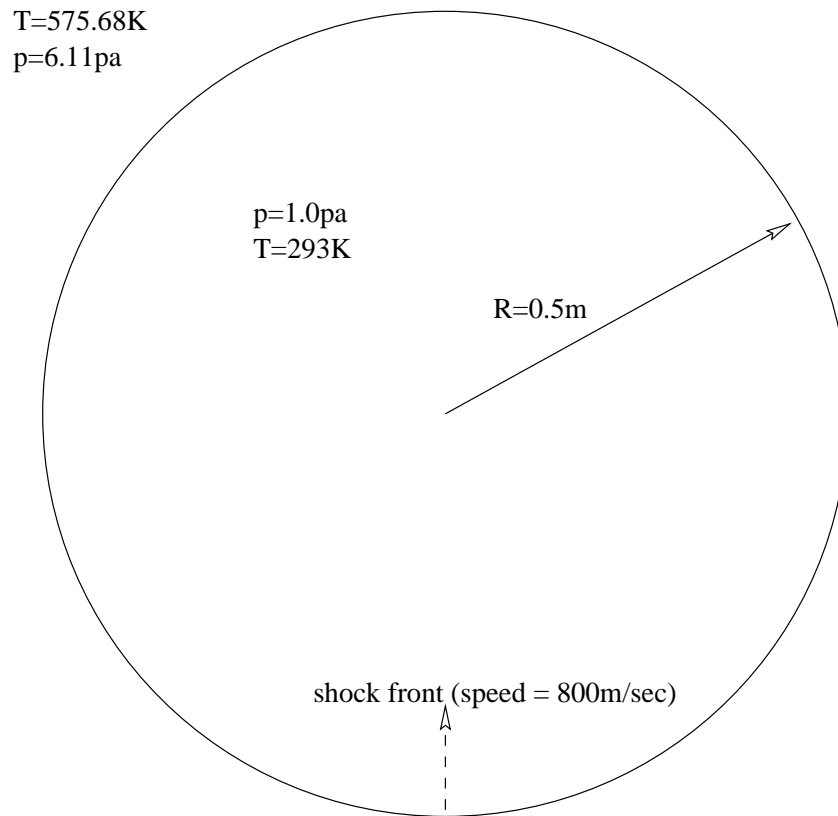


Figure 19: Initial conditions for the imploding spherical shock wave

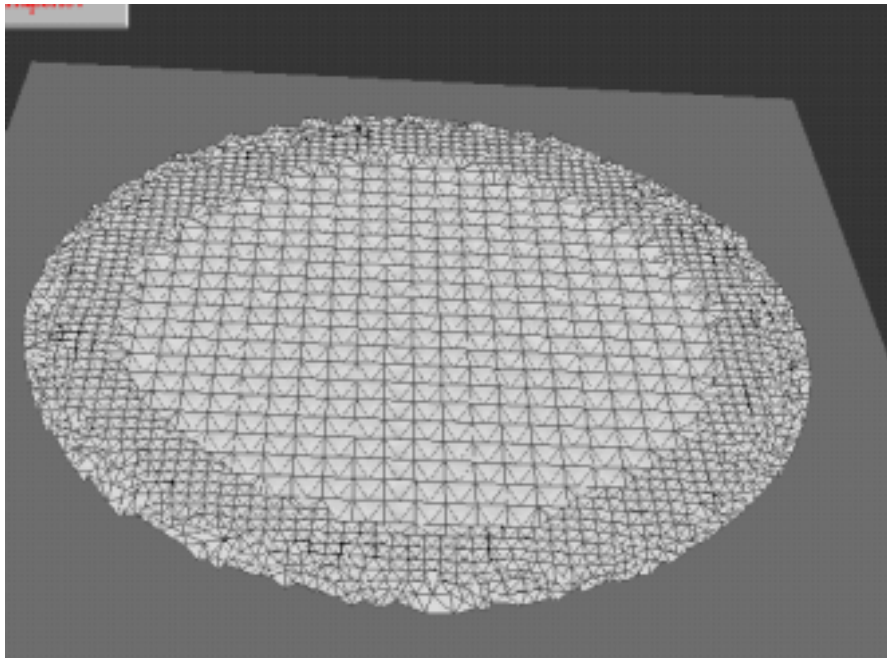


Figure 20: Adapted mesh at $t=0$

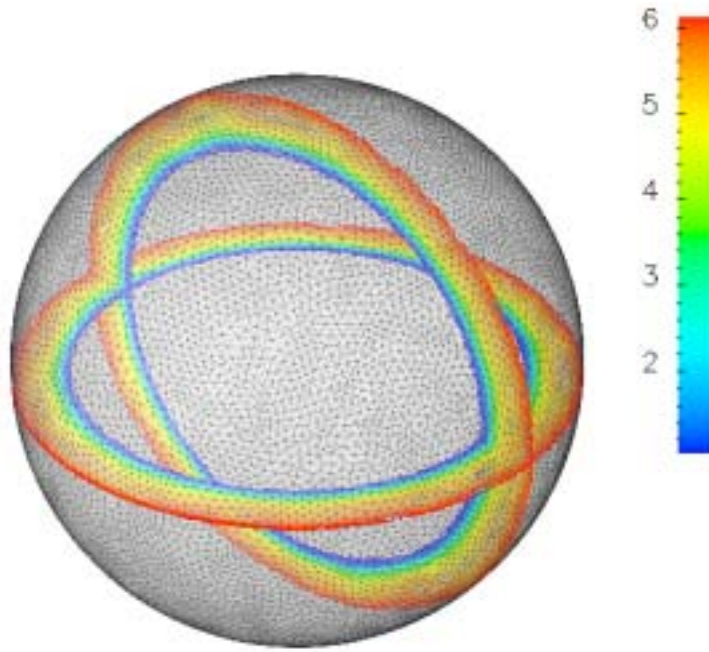


Figure 21: Spherically inward propagating shock at $t=0.1\text{sec}$

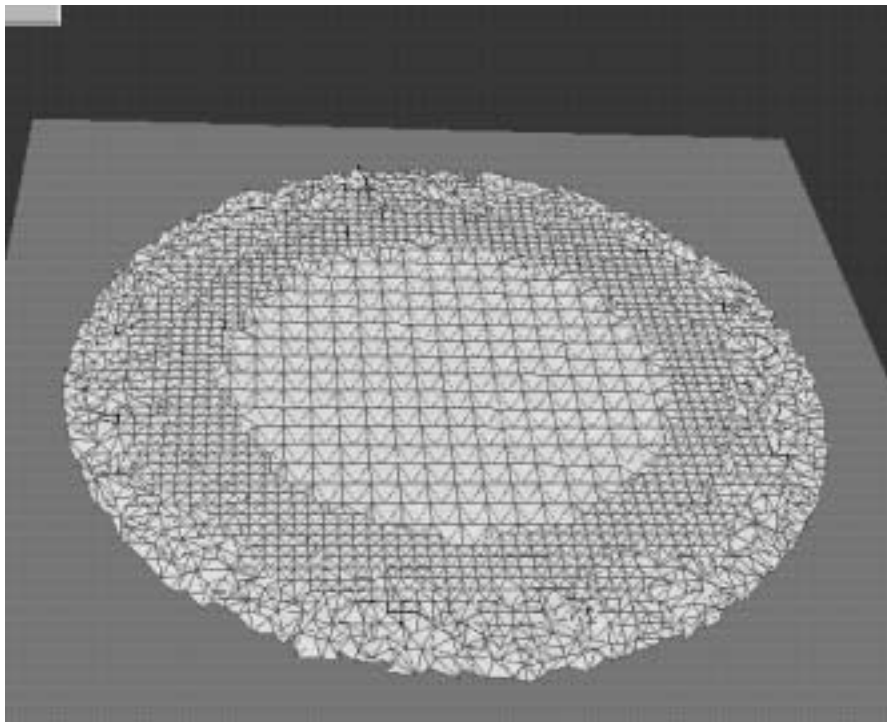


Figure 22: Mesh adapted to the shock front at $t=0.1\text{sec}$

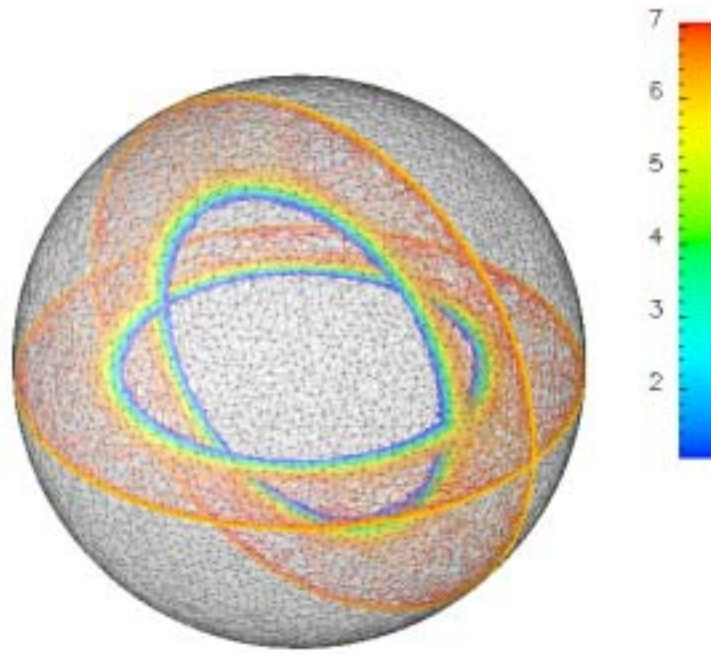


Figure 23: Shock propagating towards origin at $t=0.2\text{sec}$

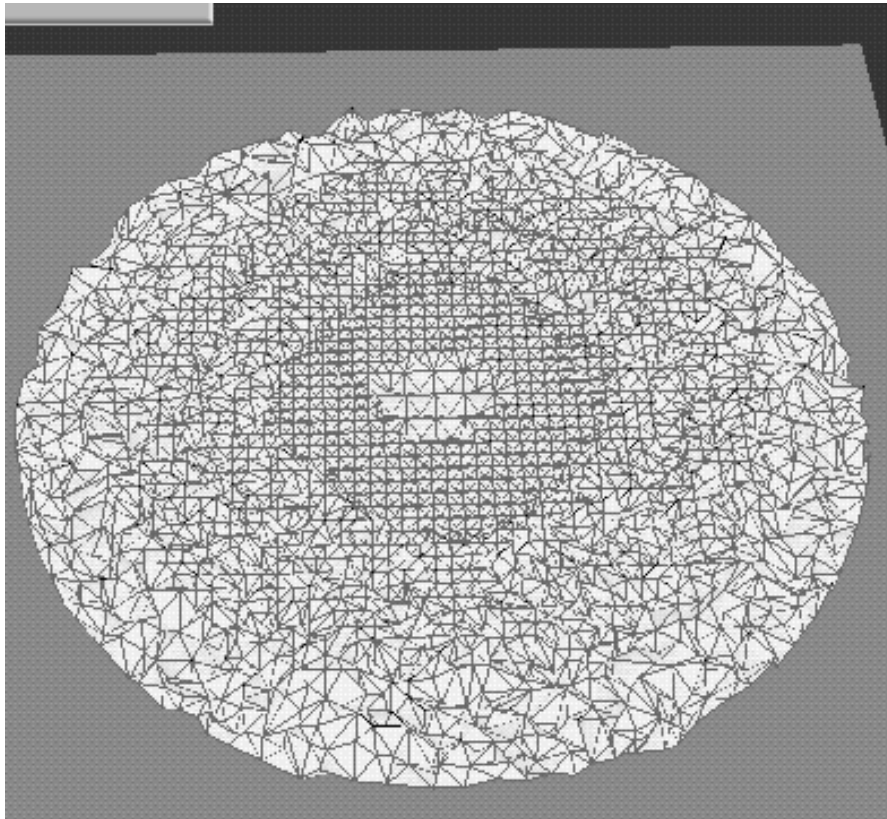


Figure 24: Adaptively refined mesh at $t=0.2\text{sec}$

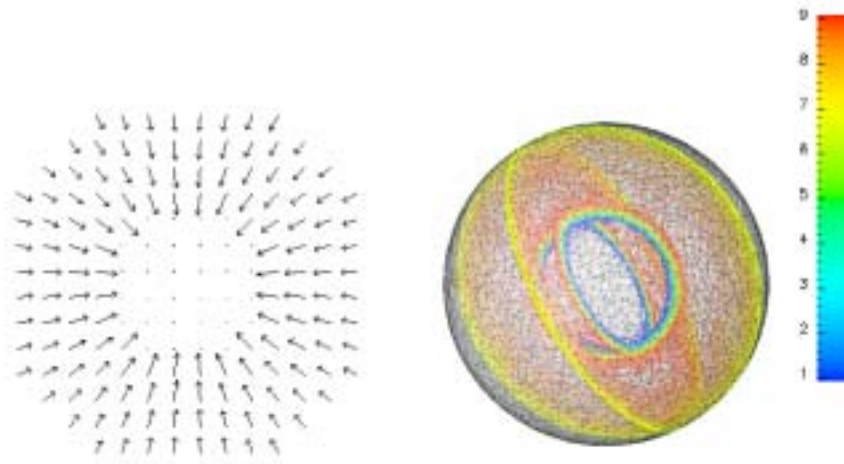


Figure 25: Velocity and pressure distributions at $t=0.3\text{sec}$

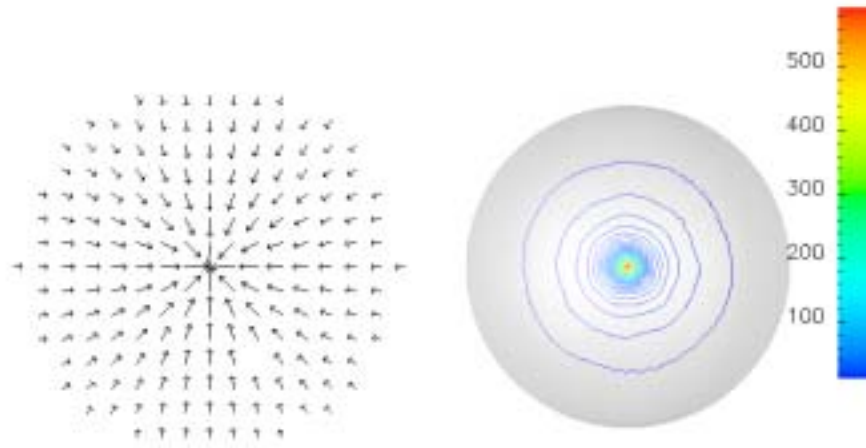


Figure 26: At the instant of reaching origin at $t=0.4680\text{sec}$

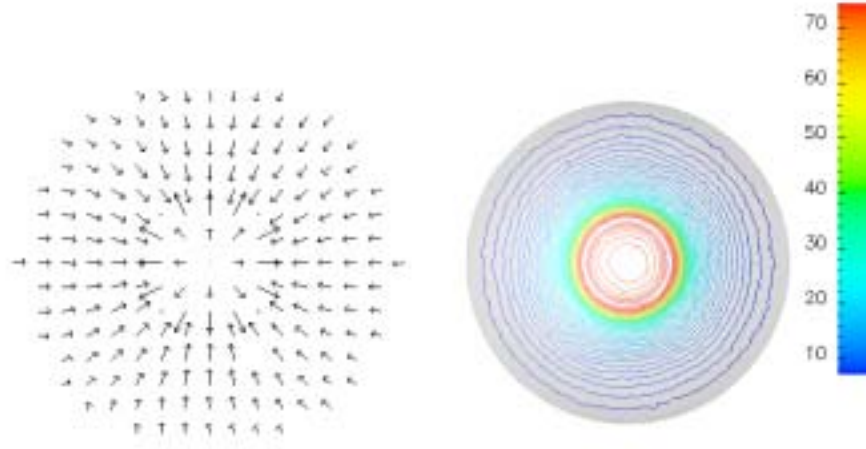


Figure 27: Shock wave reflecting back from origin at $t=0.6080\text{sec}$

moving with speed of 800m/sec , which corresponds to a Mach number of 2.3. As initially the shock is located at the boundary of the domain, the mesh is refined near the boundary as shown in Figure 20. The refined mesh consists of 715,651 regions(elements). Figure 21 shows the shock front at $t=0.1\text{ sec}$. At this point, the shock is getting strengthened slowly, as one can see from the increase in the pressure at the front. Again the mesh is regenerated by coarsening farther away at the boundary as the shock no longer exists there, Figure 22, and refining the region in the vicinity of the shock. Again the number of regions in the modified mesh is 779,946. Figure 23 presents the strengthening shock front with time at $t=0.2\text{ sec}$, and the Figure 24 presents the mesh refined at this time to further resolve the shock. The velocity and the pressure distributions at $t=0.3\text{ seconds}$ is presented in Figure 25. We can clearly see a shock accelerating towards origin. Figure 26 depicts the instant where the shock reached the origin. From the pressure distribution we can see a very high pressures is reached at the instant of shock impact. This collapse occurred after 0.4680 sec indicating the rapid time scales involved. After the collapse (i.e, impact) of the shock at the origin the shock wave bounces back and travels away from origin, as shown in Figure 27. The results predicted were compared against a self similarity solution [31], and the front speed from the simulations were found to be in good agreement.

9.2.4 Rayleigh-Taylor Instability

Another important test case was studied to verify the ability of PHASTA-2C to resolve the propagating interfaces. A Rayleigh-Taylor instability occurs when a heavy fluid pushes into a lighter one. Imagine a horizontal interface in which a heavy fluid lies above a lighter fluid, with gravity pointing downwards. The interface in this case is unstable and any small perturbations to the interface grows with time. The heavier fluid on the top pushes through these perturbations and long fingers of the heavier fluid reach down into the lighter liquid. At the same time the plumes of lighter fluid grow upward.

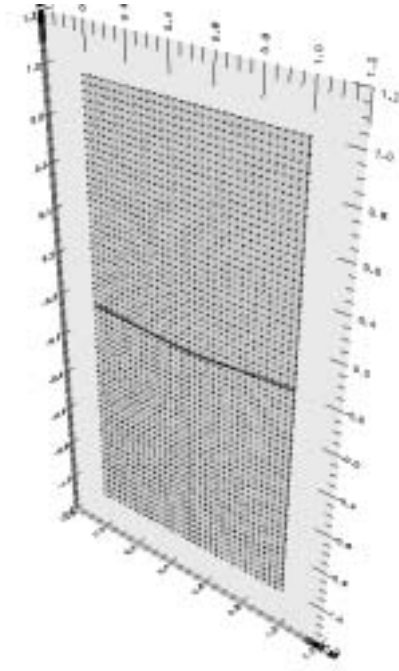


Figure 28: Initial perturbation to the interface and geometry

Simulations were done on a rectangular domain of 1m width and 2m height with walls on the upper and the lower surfaces and periodic boundary conditions in the horizontal direction. The mesh consists of $32 \times 64 \times 1$ elements. Both the fluids were chosen to be air with a density ratio of 2:1. The amplitude of the initial perturbation was 0.015m. The initial sinusoidal perturbation is shown in the Figure 28. Figure 29 shows the onset of the Rayleigh-Taylor instability. Figures 30 and 31 shows the non-linear growth of the interfacial instability. Finally it assumes the expected mushroom shape at $t=8\text{sec}$, as shown in the Figure 32.

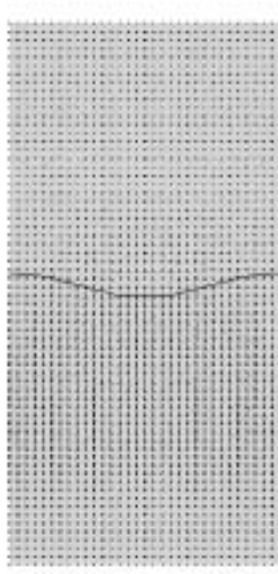


Figure 29: Onset of instability at $t=2.0\text{sec}$

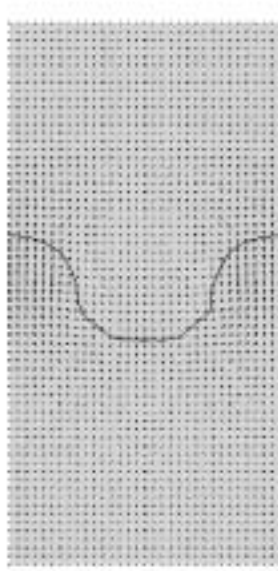


Figure 30: Further propagation of instability at $t=4.0\text{sec}$



Figure 31: Formation of Mushroom shape at $t=6.0$

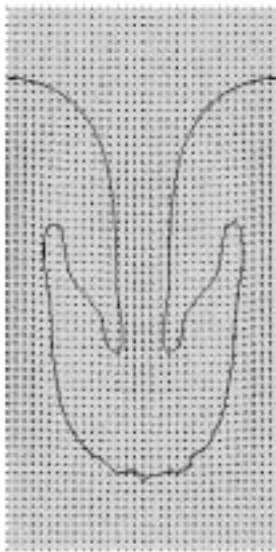


Figure 32: Further growth of instabilities at $t=8.0$

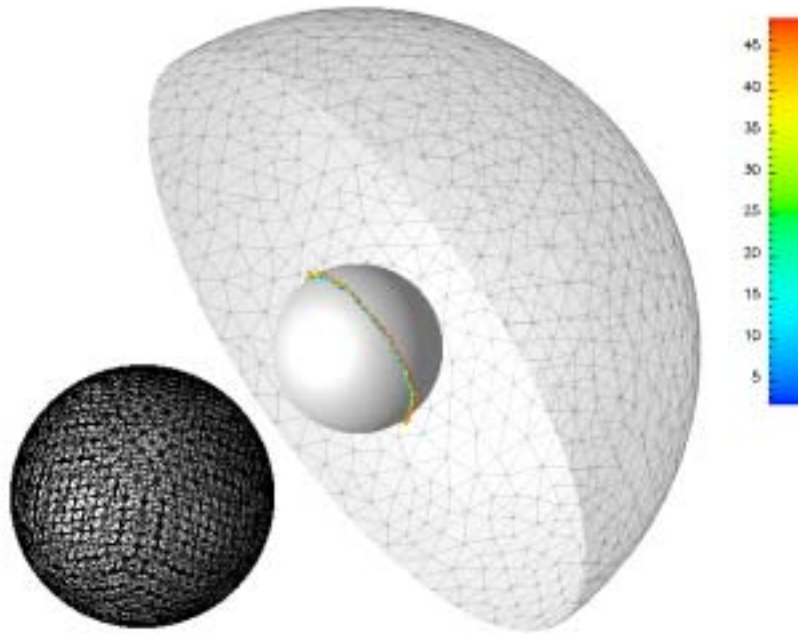


Figure 33: Imploding air bubble in water, initial conditions and mesh

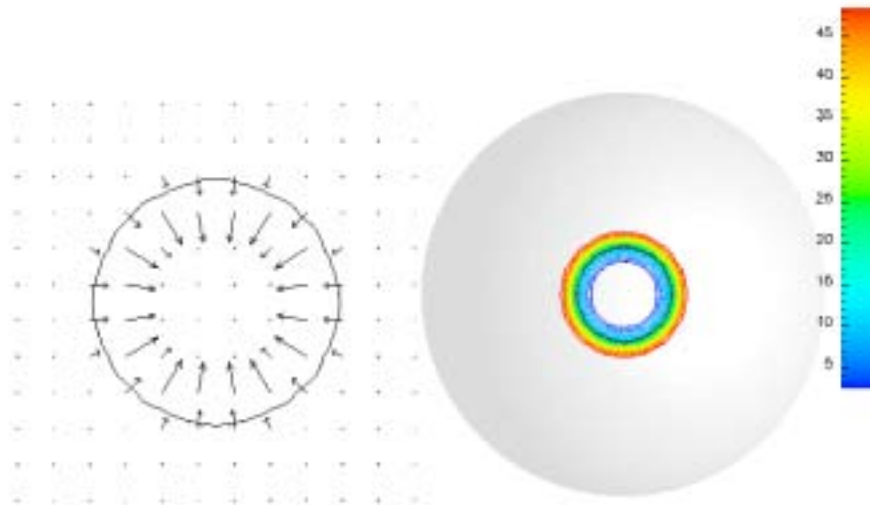


Figure 34: Early stages of implosion

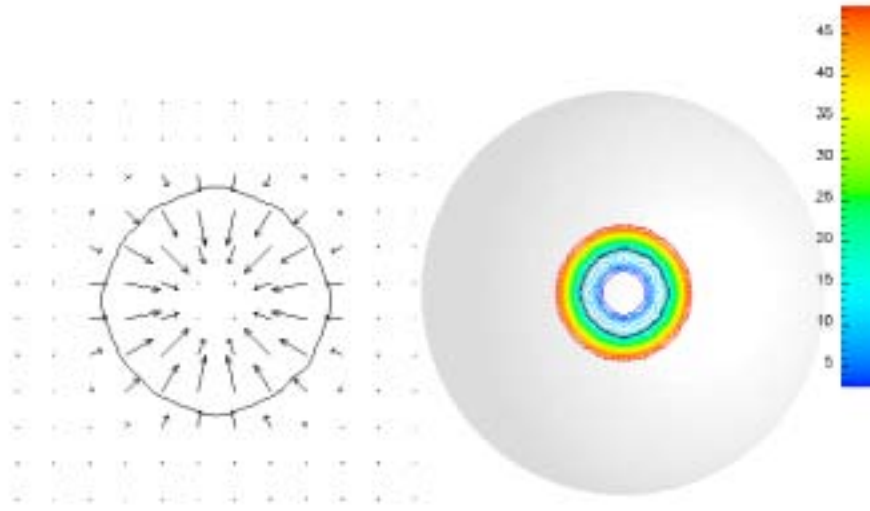


Figure 35: Inward acceleration of gas at $t=0.05\mu s$

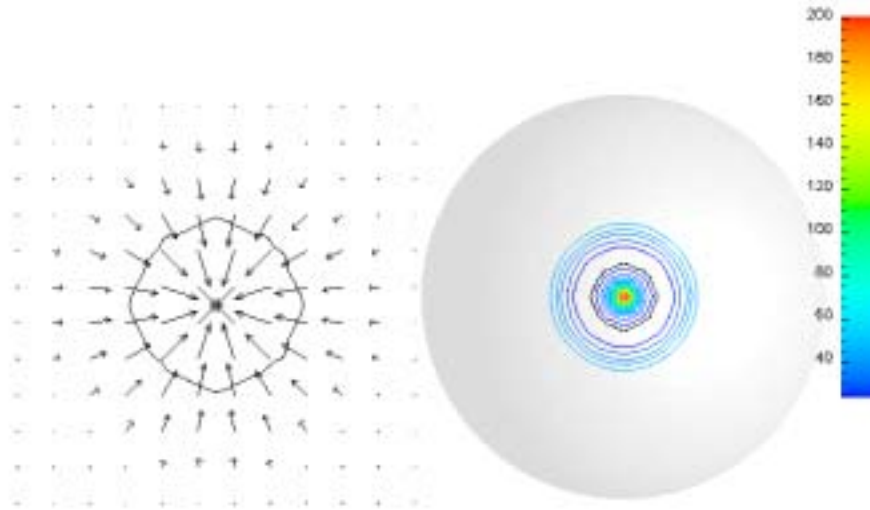


Figure 36: Collapse of the shock at center, $t=0.1540\mu s$

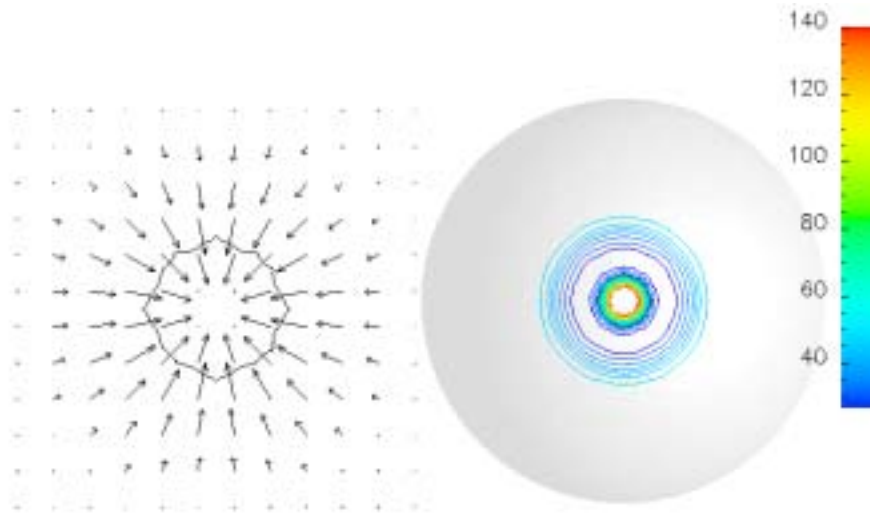


Figure 37: Rarefaction wave moving away from the center, $t=0.2\mu s$

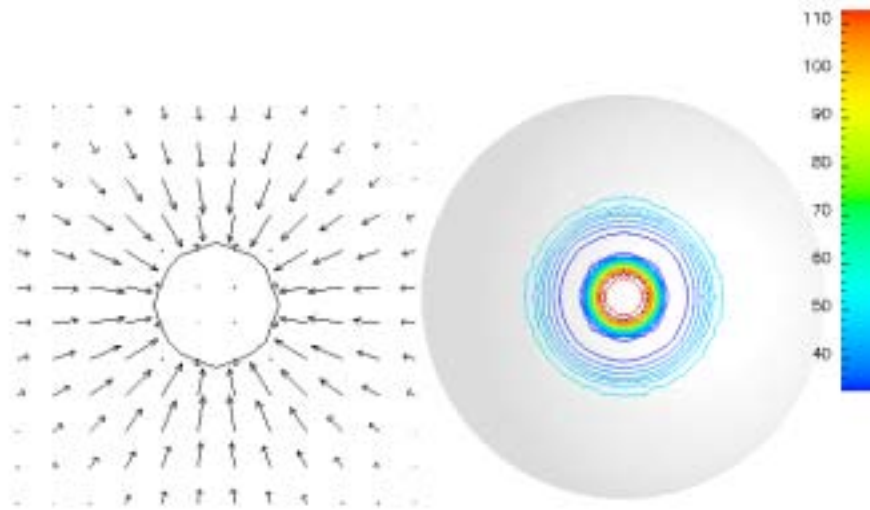


Figure 38: Shock front crossing the interface, $t=0.3\mu s$

9.2.5 Imploding Air Bubble in Water

This problem is the first step to simulate single bubble sonoluminescence (SBL). SBL involves a series of expansions and compressions of the bubble in a standing acoustic field, a problem which is similar to the one considered in this section. An air bubble at atmospheric pressure is driven by the surrounding pressurized liquid (50 atm). The initial conditions and the mesh are shown in the Figure 33. The figure shows the adaptively refined the mesh in the vicinity of the bubble to resolve the interface accurately. An early stage of implosion is shown in the Figure 34. As the bubble is compressed the gas inside the bubble gets accelerated as shown in the Figure 35. Figure 36 shows the instant of the shock collapse (i.e, impact) at $0.1540\mu\text{s}$. Figure 37 shows the rarefaction wave moving away from the center, and still at this instant the bubble is still collapsing. Once the rarefaction wave moving back crosses the interface (Figure 38), the bubble stops collapsing and starts expanding. This expansions and collapse continues. The physics predicted by the code is very much similar to the observations made in experiments. Also, these simulations indicate that neither Taylor, or Farady instabilities lead to bubble break up. This implies the 1-D simulations of bubble fusion should be realistic.

10 The Analysis of Bubble Fusion

Bubble Fusion may occur if the density and temperature during an implosion are large enough to induce thermonuclear fusion for solutions in which there is sufficient deuterium (D) or tritium (T) in the highly compressed vapor. Bubble Fusion has been recently reported by researchers at ORNL and RPI [32] in cavitation bubbles formed in deuterated acetone (C_3D_6O). Simulations were made to more fully understand these data. A brief description of the model and simulatios is as follows. The Mi-Gruneisen equation of state for a highly compressed [31], [33] was used, in particular:

$$\begin{aligned}\epsilon &= \frac{e}{p} = \epsilon_p + \epsilon_T, \\ p &= p_p + p_T, \\ \rho_T &= \rho\Gamma(\rho, T)\bar{c}_v T, \\ \epsilon_T &= \bar{c}_v(\rho, T)T\end{aligned}\tag{51}$$

where ϵ_p and p_p are the potential, are the potential, or “cold,” components, and ϵ_T and p_T are the thermal components of the internal energy and pressure, respectively; Γ is the Gruneisen coefficient, and \bar{c}_v is the average heat capacity at constant volume, and T is the absolute temperature of each phase.

The potential, or “cold,” components characterize intermolecular force interactions which depend on the average distances between the molecules, and thus depend on the density, ρ . For rarefied gases, where these distances are very large (i.e., for small densities, ρ), the potential components are negligibly small. In contrast, the potential components (ϵ_p and p_p) are essential for dense gases (i.e., at high pressure) and for condensed (liquid and solid) states of matter. The thermal components (ϵ_T and p_T) characterize the internal energy and pressure due to thermal (chaotic) motion of the molecules.

For most fluids, the potential components can be represented by a Born-Mayer potential [31], [33]:

$$p_p = A\left(\frac{\rho}{\rho_0}\right)^{2/3} \exp[b(1 - (\frac{\rho_0}{\rho})^{1/3})] - K\left(\frac{\rho}{\rho_0}\right)^{n+1} + \Delta p_p, \quad (52)$$

$$\epsilon_p = \frac{3A}{\rho_0 b} \exp[b(1 - (\frac{\rho_0}{\rho})^{1/3})] - \frac{3K}{\rho_0} \left(\frac{\rho}{\rho_0}\right)^n + \Delta \epsilon_p \quad (53)$$

where, A , K , b , and n are constant coefficients [32], which completely specify the Born-Mayer potential, and $\Delta \epsilon_p$ is a correction for potential energy. We note that the different phases (e.g., liquid and vapor) take place only for subcritical conditions. That is, for, $p < p_{cr}$, $T < T_{cr}$.

To simulate electron conductivity in an ionized gas, we assume,

$$k_g = a_{1g} T^m + a_{2g}, \quad (54)$$

where a_{1g} , a_{2g} and m are appropriate constants and T is the effective temperature of the plasma.

At very high temperatures, where ionization take place, we may use [34]:

$$\epsilon_T = \bar{c}_v T(1 - \beta_D) + \beta_D[(1 - \beta_I)(c_{v_D} T + \epsilon_D) + \beta_I(c_{v_I} T + \epsilon_I)] \quad (55)$$

$$p_T = \rho \Gamma \epsilon_T \quad (56)$$

$$\Gamma = \Gamma(\rho)(1 - \beta_D) + \beta_D \Gamma_D \quad (57)$$

$$\beta_k = \exp(-(\frac{T_k}{2T})^2), (k = D, I) \quad (58)$$

where [34], $\epsilon_D = R_g T_D / \bar{M}$, $\epsilon_I = 2R_g T_I / \bar{M}$, \bar{M} is the molecular weight, and the subscripts “D” and “I” stand for disassociated ionized fluid, respectively.

These equations were evaluated for the special case of spherically symmetric bubble dynamics, using the following interfacial jump conditions:

$$T_l|_{r=a} - T_g|_{r=a} \equiv [T] = 0.45 \frac{m'' T_{sat}}{\sqrt{2R_g T_{sat} \rho|_{r=a}}} \quad (59)$$

$$k_l \frac{\partial T_l}{\partial r}|_{r=a} - k_g \frac{\partial T_g}{\partial t}|_{r=a} = m'' h_{fg}(p_g|_{r=a}) \quad (60)$$

where, $r = a(t)$ is the location of the bubble's interface (i.e., $a \equiv R$).

Using the Hertz-Knudsen-Langmuir model [35], the phase change mass flux is:

$$m'' = \frac{\alpha}{\sqrt{2\pi R_g}} \left(\frac{p_{sat}(T_l|_{r=a})}{\sqrt{T_l|_{r=a}}} - \frac{\chi p_g|_{r=a}}{\sqrt{T_g|_{r=a}}} \right) \quad (61)$$

where,

$$\chi = \exp(-\Omega^2) - \Omega \sqrt{\pi} (1 - \frac{2}{\sqrt{\pi}} \int_0^\Omega \exp(-x^2) dx), \Omega = \frac{m''}{\sqrt{2p_g|_{r=a}}} \sqrt{R_g T_g|_{r=a}}, \quad (62)$$

and α is the phase change (i.e., accommodation) coefficient.

In order to estimate the production of fusion neutrons, the neutron kinetics model given by Gross [1984] were used in conjunction with the HYDRO code evaluations of the local bubble implosion thermal-hydraulics. This model uses the weighted cross sections of the fusion reactions which implicitly assume that the plasma has an equilibrium distribution and for D/D fusion reactions is given by:

$$\dot{n}_n''' \equiv \frac{dn_n'''}{dt} = \frac{1}{2}(\overline{\sigma v})_{D/D}(n_D''')^2 \quad (63)$$

while for D/T fusion reactions we have:

$$\dot{n}_n''' \equiv \frac{dn_n'''}{dt} = (\overline{\sigma v})_{D/T}(n_D''')(n_T''') \quad (64)$$

where, n_n''' is the concentration of fusion neutrons produced (eg, neutrons/m³), n_h''' is the concentration of the hydrogen isotope in question (eg, deuterium, D, or tritium, T), and $(\overline{\sigma v})$ are the weighted cross sections. Typical numerical results for D-acetone during the low Mach number stage of bubble dynamics in the ORNL experiments [32] are shown in Figure 39. It can be seen that shock-induced intensification during bubble cloud collapse [32] was assumed, and the bubble radius and vapor mass are sensitive to the impressed pressure. Similar runs have shown that bubble compression is also very sensitive to the temperature of the liquid pool, T_0 . Interestingly, because the phase change coefficient for heavy water, α , is relatively low (~ 0.075) a significant amount of vapor remains during the implosion process, cushioning the collapse and mitigating the compression. Fortunately, D-acetone has a large value of α (~ 1.0), thus it is a much better test fluid. Figures 40 & 41 show the evaluations of typical ORNL D-acetone test conditions for two different acoustic forcing pressures. It can be seen that the peak pressure, temperature and density are large. As a result, as can be seen in Figure 41, about 1.4 neutrons are produced per implosion, which for the forcing frequency (20kHz) used in the ORNL experiments, implies a neutron production rate of about 3×10^4 n/s, which is in good agreement with what was actually measured [32]. While this is not a large neutron intensity, it should be noted that the ORNL experiments only had one pressure antinode, and no attempt was made to optimize neutron production. In any event, it appears that HYDRO codes, like the PHASTA-2C code, are capable of simulating Bubble Fusion phenomena.

It should be noted that, multidimensional phenomena appear to be the key to the possible development of a nuclear power reactors based on Bubble Fusion. There is significant pressure intensification within and imploding bubble cloud, and this may lead to the combinations required for Bubble Fusion [32]. Moreover, in a typical manifestation of a bubble

fusion nuclear reactor, there might be an acoustically forced deuterated-liquid-filled chamber in which there are multiple antinodes in both the lateral and axial directions. Those antinodes at low pressure might experience neutron-induced bubble nucleation, as a consequence of the antinodes at high pressure in which there is bubble implosion and neutron production. This suggests the possibility of a self-sustaining chain reaction. Nevertheless, to determine the feasibility, stability and efficiency of this concept, one must analyze the interactions within and between the various bubble clusters formed. A detailed 3-D simulation is essential, and, to the best of our knowledge, PHASTA-2C is the only code in the open literature which has these capabilities.

11 Conclusions

A state-of-the-art HYDRO code, PHASTA-2C, has been developed. The presented research clearly demonstrate the following achievements

1. Implementation of the level set method in an adaptive mesh FEM code for solving multi-phase flow problems
2. Successfully handled the multiphase flows with rapidly changing interfaces
3. Employed the discontinuity capturing techniques for solving flows with strong shocks
4. Demonstrated the ability to predict the compressible gas dynamics
5. Novel applications of the Ghost Fluid method
6. Efficient computations using the adaptive mesh strategy
7. Validated available results in literature
8. The ability to calculate strong bubble implosions, such as, those which may lead to ultra high temperatures and pressures

The preliminary studies clearly illustrated the capability of PHASTA-2C to predict both single and multiple bubble phenomena including: sonoluminescence, sonochemistry and Bubble Fusion.

12 Acknowledgements

The helpful and the stimulating discussions held with Dr.R.P. Taleyarkhan, Dr.I.K. Akhatov and Dr.R.I. Nigmatulin are greatly acknowledged.

References

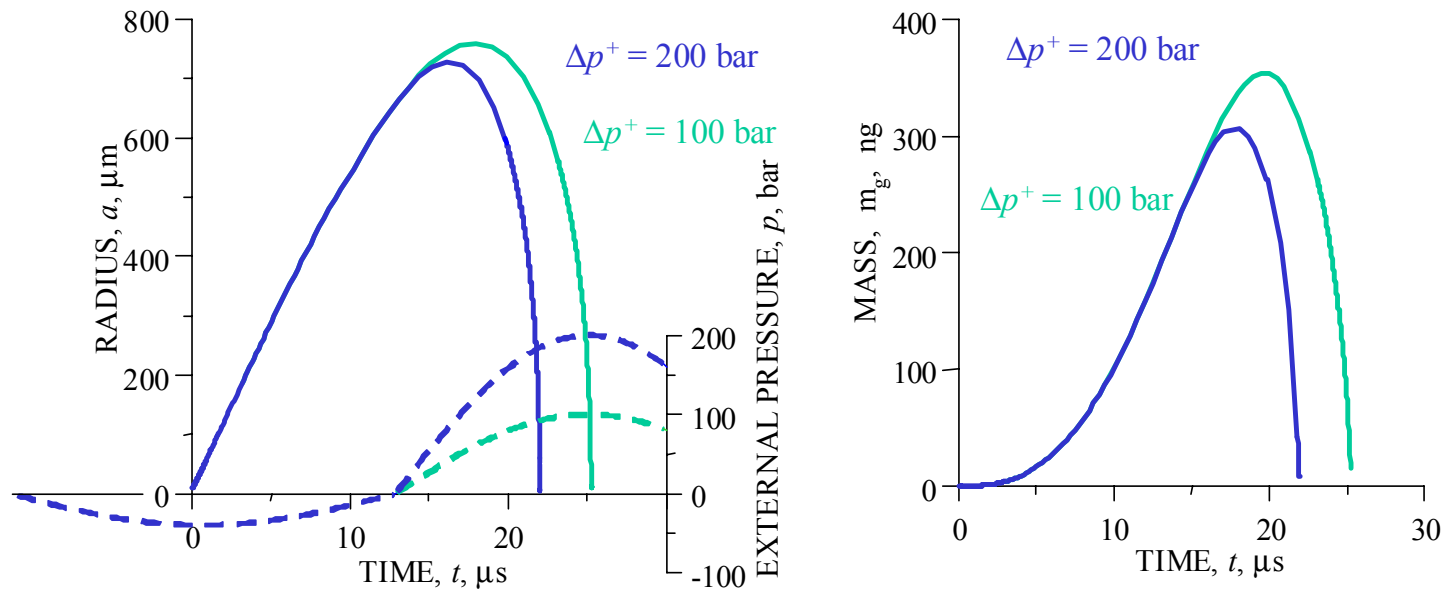
- [1] S. O. Unverdi and G. Tryggvason, “A front tracking method for viscous, incompressible, multifluid flows”, *Journal of Computational Physics*, **100** (1992) 25.
- [2] J. P. Best, “The formation of toroidal bubbles upon the collapse of transient cavities”, *Journal of Fluid Mechanics*, **251** (1993) 79.
- [3] C. W. Hirt and B. D. Nichols, “Volume of fluid (VOF) method for the dynamics of free boundaries”, *Journal of Computational Physics*, **39** (1981) 201.
- [4] J. A. Sethian, *Level Set Methods and Fast Marching Methods*. Cambridge U. Press, 1999.
- [5] M. Sussman, P. Smereka, and S. J. Osher, “A level set approach for computing solutions to incompressible two-phase flows”, *Journal of Computational Physics*, **114** (1994) 146.
- [6] M. Sussman, J. B. Bell A. S. Almgren, L. H. Howell P. Colella, and W. L. Welcome, “An adaptive level set approach for incompressible two-phase flows”, *Journal of Computational Physics*, **148** (1999) 81–124.
- [7] T. J. R. Hughes and M. Mallet, “A new finite element formulation for fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems”, *Comp. Meth. Appl. Mech. Engng.*, **58** (1986) 305–328.

- [8] T. J. R. Hughes, L. P. Franca, and M. Mallet, “A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics”, *Comp. Meth. Appl. Mech. Engng.*, **54** (1986) 223–234.
- [9] T. J. R. Hughes, L. P. Franca, and M. Mallet, “A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems.”, *Comp. Meth. Appl. Mech. Engng.*, **58** (1986) 329–336.
- [10] J. U. Brackbill, D. B. Kothe, and C. Zemach, “A continuum method for modeling surface tension”, *Journal of Computational Physics*, **100** (1992) 335–354.
- [11] K.E. Jansen, “Large-eddy simulation using unstructured grids”, in C. Liu and Z. Liu, editors, *Advances in DNS/LES*, Greyden Press, Columbus, Ohio, (1997) 117–128.
- [12] K. E. Jansen, “A stabilized finite element method for computing turbulence”, *Comp. Meth. Appl. Mech. Engng.*, **174** (1999) 299–317.
- [13] G. Hauke, *A unified approach to compressible and incompressible flows and a new entropy-consistent formulation of the k - ϵ model*, Ph.D. thesis, Stanford University, 1995.
- [14] G. Hauke and T. J. R. Hughes, “A comparative study of different sets of variables for solving compressible and incompressible flows”, *Comp. Meth. Appl. Mech. Engng.*, **153** (1998) 1–44.
- [15] A. N. Brooks and T. J. R. Hughes, “Streamline upwind / Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations”, *Comp. Meth. Appl. Mech. Engng.*, **32** (1982) 199–259.
- [16] F. Shakib, *Finite element analysis of the compressible Euler and Navier-Stokes equations*, Ph.D. thesis, Stanford University, 1989.
- [17] L. P. Franca and S. Frey, “Stabilized finite element methods: II. The incompressible Navier-Stokes equations”, *Comp. Meth. Appl. Mech. Engng.*, **99** (1992) 209–233.

- [18] G. Hauke and T. J. R. Hughes, “A unified approach to compressible and incompressible flows”, *Comp. Meth. Appl. Mech. Engng.*, **113** (1994) 389–396.
- [19] C. H. Whiting and K. E. Jansen, “A stabilized finite element method for the incompressible Navier-Stokes equations using a hierarchical basis”, *International Journal of Numerical Methods in Fluids*, **35** (2001) 93–116.
- [20] K. E. Jansen, C. H. Whiting, and G. M. Hulbert, “A generalized- α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method”, *Comp. Meth. Appl. Mech. Engng.*, **190** (1999) 305–319.
- [21] T. J. R. Hughes, *The finite element method: Linear static and dynamic finite element analysis*. Prentice Hall, Englewood Cliffs, NJ, 1987.
- [22] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, “Eulerian capturing methods based on a level set formulation for incompressible fluid interfaces”, *Journal of Computational Physics*, **124** (1996) 449.
- [23] Ronald P. Fedkiw, Tariq Aslam, Barry Merriman, and Stanely Osher, “A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method)”, *Journal of Computational Physics*, **152** (1999) 457–492.
- [24] D. Benson, “Computational methods in lagrangian and eulerian hydrocodes”, *Comp. Meth. Appl. Mech. Engng.*, **99** (1992) 235.
- [25] Ronald P. Fedkiw, Barry Merriman, and Stanely Osher, *Numerical methods for a one-dimensional interface separating coimpressible and incompressible flows, in Barriers and Challenges in Computational Fluid Dynamics*. Cambridge U. Press, 1999.
- [26] J.-P. Cocchi and Saurel, “A riemann problem based method for the resoltuion of compressible multimaterial flows”, *Journal of Computational Physics*, **137** (1997) 265.
- [27] J. R. Grace, “Shapes and velocities of bubbles rising in infite liquids”, *Transactions of Institution of Chemical Engineers*, **51** (1973) 116–120.

- [28] E. Delnoij, J. A. M. Kuipers, and W. P. M. van Swaaij, “Computational fluid dynamics applied to gas-liquid contactors”, *Chemical Engineering Science*, **52(21/22)** (1997) 3623.
- [29] J. K. Walters and J. F. Davidson, “The initial motion of a gas bubble formed in an inviscid liquid. part-2. the three dimensional bubble and the toroidal bubble”, *Journal of Fluid Mechanics*, **17** (1963) 321–336.
- [30] Gary A. Sod, “A survey of finite difference methods for systems of nonlinear hyperbolic conservation laws”, *Journal of Computational Physics*, **27** (1978) 1–31.
- [31] Y. B. Zeldovich and Y. P. Raizer, *Physics of Shock Waves and High Temperature Hydrodynamics Phenomena*. Academic Press, 1966.
- [32] R. P. Taleyarkhan, C. D. West, J.S. Cho, Jr R. T. Lahey, R. I. Nigmatulin, and R. C. Block, “Evidence for nuclear emissions during acoustic cavitation”, *Science*, **295** (2002) 1868–1873.
- [33] R. I. Nigmatulin, *Dynamics of Multiphase Media*. Hemisphere, New York, 1991.
- [34] W. C. Moss, D. B. Clarke, J. W. White, and D. A. Young, “Hydrodynamic simulations of bubble collapse and picosecond sonoluminescence”, *Physics of Fluids*, **6** (1994) 2979.
- [35] R. W. Schrage, *A Theoretical Study of Interphase Mass Transfer*. Columbia Univ. Press, New York, 1953.

Vapor Bubble Collapse in Deuterated Acetone, $\text{C}_3\text{D}_6\text{O}$ (Low Mach Number Stage)

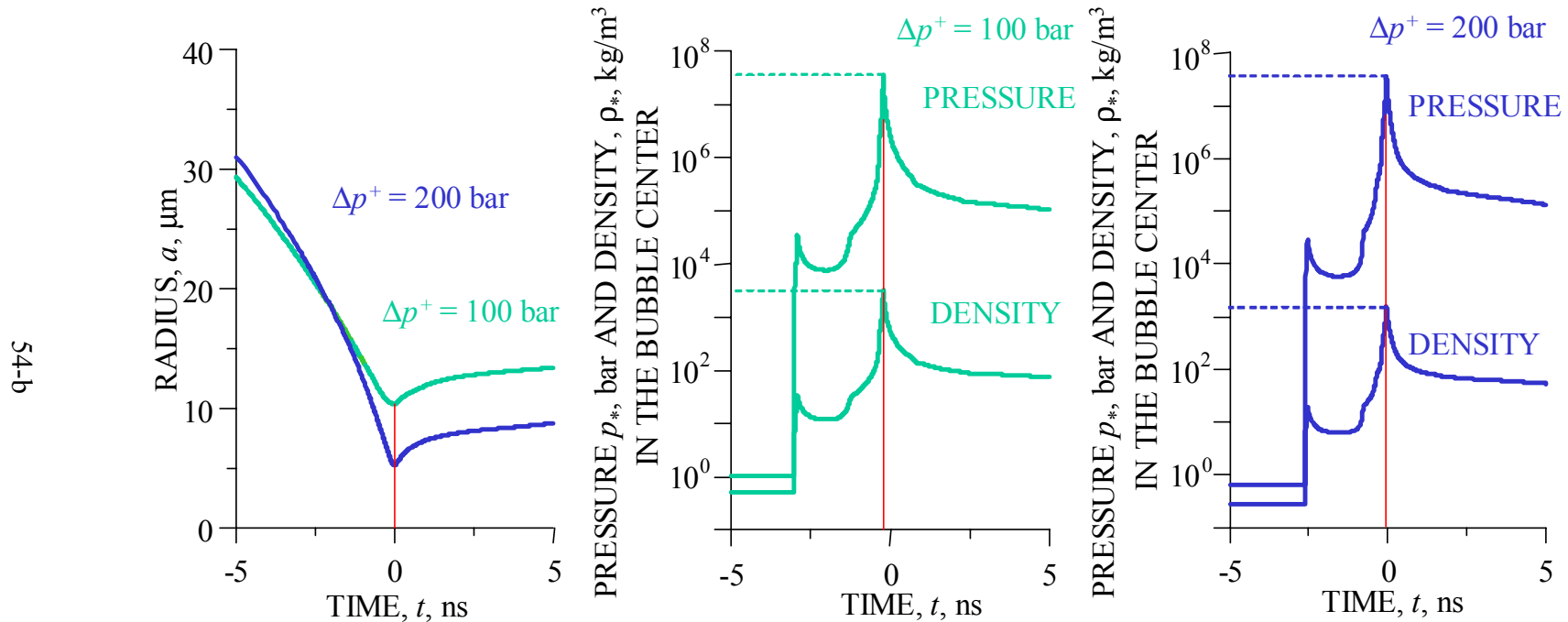


Bubble radius and vapor mass evolution at various amplitudes, Δp ,
of driving pressure.

$$f = 20.5 \text{ kHz}, \quad T_0 = 273 \text{ K}, \quad \alpha = 1.0.$$

Figure 39

Vapor Bubble Collapse in Deuterated Acetone, C_3D_6O (High Mach Number Stage)

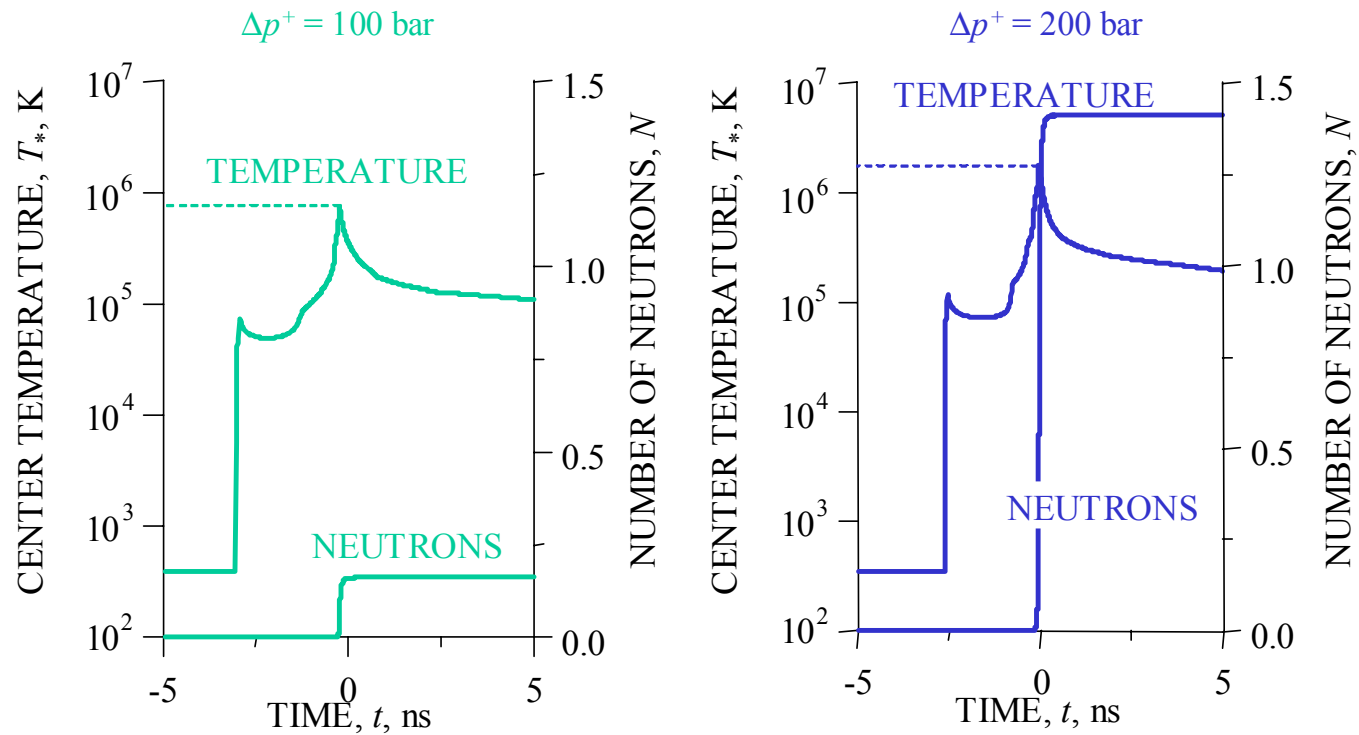


Density and pressure evolution of acetone in the center of the bubble.

$$f = 20.5 \text{ kHz}, \quad T_0 = 273 \text{ K}, \quad \alpha = 1.0.$$

Figure 40

Vapor Bubble Collapse in Deuterated Acetone, C_3D_6O (High Mach Number Stage)



Temperature in the center of the bubble, T_*
and number of emitted neutrons, N as functions of time.
 $f = 20.5$ kHz, $T_0 = 273$ K, $\alpha = 1.0$.

Figure 41

